

© Copyright 1992

Charles Teorell Loop

Generalized B-spline Surfaces
of
Arbitrary Topological Type

by

Charles Teorell Loop

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

University of Washington

1992

Doctoral Dissertation

In presenting this dissertation in partial fulfillment of the requirements for the Doctoral degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of this dissertation is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for copying or reproduction of this dissertation may be referred to University Microfilms, 300 North Zeeb Road, Ann Arbor Michigan 48106, to whom the author has granted "the right to reproduce and sell (a) copies of the manuscript in microform and/or (b) printed copies of the manuscript made from microform."

Signature_____

Date_____

University of Washington

Abstract

Generalized B-spline Surfaces
of
Arbitrary Topological Type

by Charles Teorell Loop

Chairperson of the Supervisory Committee: Professor Tony DeRose

Department of Computer Science
and Engineering

B-spline surfaces, although widely used, are incapable of describing surfaces of arbitrary topological type. It is not possible to model a general closed surface or a surface with handles as a single non-degenerate B-spline. In practice such surfaces are often needed. In this thesis, a generalization of bicubic tensor product and quartic triangular B-spline surfaces is presented that is capable of capturing surfaces of arbitrary topological type. These results are obtained by relaxing the sufficient but not necessary smoothness constraints imposed by B-splines and through the use of an n -sided generalization of Bézier surfaces called S-patches.

Table of Contents

List of Figures	iv
List of Tables	vi
Chapter 1: Introduction	1
1.1 Parametric Surfaces	2
1.2 Implicit Surfaces	6
1.3 Subdivision Surfaces	7
1.4 Parametric Surfaces of Arbitrary Topological Type	9
1.5 Generalized B-splines	12
1.6 Overview	14
Chapter 2: Bézier Forms	15
2.1 Preliminary Definitions	15
2.2 Bézier Simplices	16
2.2.1 Polar Forms	17
2.2.2 DeCasteljau's Algorithm	18
2.2.3 Derivatives of Bézier Simplices	19
2.2.4 Degree Elevation	19
2.3 Bézier Curves	19
2.3.1 The Derivative of a Bézier Curve	20
2.4 Tensor Product Bézier Surfaces	21

2.4.1	Derivatives of Tensor Product Bézier Surfaces	22
2.5	Bézier Triangles	22
Chapter 3:	B-splines	24
3.1	B-spline Curves	24
3.1.1	B-spline Curves in Bézier Form	26
3.2	Tensor Product B-spline Surfaces	28
3.2.1	Tensor Product B-spline Surfaces in Bézier Form	29
3.3	Triangular B-spline Surfaces	31
3.3.1	Triangular B-splines in Bézier Form	32
Chapter 4:	S-patches	34
4.1	S-patch Basics	34
4.2	Bézier Surfaces are Generalized	37
4.3	Representing Polynomials	38
4.4	Joining S-patches Smoothly	39
Chapter 5:	The Generalized B-spline Scheme	44
5.1	Input and Output	44
5.2	Overview of the Scheme	46
5.3	Vertex Behavior	47
5.3.1	Vertex Maps in Bézier Triangular Form	50
5.3.2	Constructing \mathbf{b}_{111}	51
5.3.3	Constructing \mathbf{b}_{300}	53
5.3.4	Constructing \mathbf{b}_{210}	53
5.3.5	Constructing \mathbf{b}_{120}	55
5.4	Edge Behavior	56
5.4.1	Boundary Curves	58
5.4.2	Boundary Differential	59
5.5	Face Behavior	61

5.5.1	Determining the Functions μ and ν	62
5.5.2	Boundary Panels	68
5.5.3	Interior Points	72
5.6	Summary	74
Chapter 6:	Specializations	81
6.1	Degree Reductions	82
6.2	Generalizing B-spline Surfaces	84
6.2.1	Bicubic B-spline Surfaces	85
6.2.2	Quartic Triangular B-spline Surfaces	91
6.3	Other Special Cases	96
Chapter 7:	Conclusions	98
Bibliography	101

List of Figures

1.1	Examples of a) a Bézier curve, b) a Bézier tensor product surface, and c) a Bézier triangle	4
1.2	A B-spline curve	5
1.3	A subdivision surface generated by the Catmull & Clark algorithm	8
2.1	A cubic Bézier curve.	20
2.2	A bicubic tensor product Bézier patch	21
2.3	A quartic Bézier triangle	23
3.1	a) A de Boor polygon b) The corresponding Bézier control points c) The resulting curve	27
3.2	a) A rectangular de Boor net b) The corresponding tensor product Bézier control nets c) The resulting surface	30
3.3	a) A triangular de Boor net b) The corresponding triangular Bézier control nets c) The resulting surface	32
4.1	Schematic representation of an S-patch.	35
4.2	The geometry of functionals $\alpha_1(p), \dots, \alpha_n(p)$	36
4.3	A depiction of the setup assumed in the definition of G^1 continuity.	40
5.1	Vertex map F	48
5.2	The construction of keypoints $\mathbf{b}_{1,111}, \dots, \mathbf{b}_{n,111}$ on a single face	52
5.3	The construction of the point \mathbf{b}_{300}	54

5.4	Edge map G	57
5.5	The boundary panels of S -patch H	72
5.6	The classic "suitcase corner" problem	76
5.7	A 5-sided patch in an otherwise rectangular control mesh	77
5.8	A closed surface with a handle	78
5.9	A closed branching surface	79
5.10	A surface with an irregular control net	80
6.1	The labeling of a tensor product B-spline control mesh	87
6.2	The labeling of a triangular B-spline control mesh	92

List of Tables

6.1	Reductions that occur for various mesh restrictions	97
7.1	The number of S-patch control points for a given number of sides (down) and depth (across)	100

Acknowledgements

I wish to express my sincerest thanks to Tony DeRose for his encouragement, patience, keen insights, and seemingly unconditional support. I would also like to thank Ron Goldman and Werner Stuetzle.

Dedication

To the memory of my father

John Wickwire Loop

Chapter 1

Introduction

This thesis is concerned with the problem of generating a mathematical description of smooth three dimensional shapes for use in Computer Aided Geometric Design (CAGD). Such shapes are useful in the design and representation of the outer skin of a car, the hull of a ship, the fuselage of an airplane, or any object that possesses a ‘free form’ shape. Parametric surfaces are an excellent tool for representing such smoothly varying sculptured objects, and B-spline techniques provide an intuitive user interface for working with parametric surfaces. However, the current theory of B-splines has serious shortcomings when modeling general closed surfaces, or surfaces with handles. In this thesis a new modeling technique is presented that is well suited to creating smooth free form surfaces of arbitrary topological type.

The scheme proposed here is closely related to B-spline surface techniques, yet does not limit the topological type of surfaces it can be used to create. In certain special cases the new scheme generalizes two types of well known B-spline surfaces: bicubic tensor product B-splines and quartic triangular B-splines. Before providing details of these claims, a review of surface design methods commonly used in Computer Graphics and CAGD is presented.

1.1 Parametric Surfaces

Parametric surfaces are the most popular method of generating free-form shapes. A parametric surface is defined by

$$f(u, v) = \{x(u, v), y(u, v), z(u, v)\},$$

where u and v are the coordinates of a point in a *domain plane*, and x, y , and z are coordinate functions that define a point in 3-space. These coordinate functions are generally polynomial or rational. One reason for the popularity of parametric surfaces is that they are easy to generate for computer display by evaluating the coordinate functions of the surface at various locations in the domain plane. It is also easy to find surface normals and curvatures by evaluating the derivatives of the coordinate functions.

Parametric surfaces in CAGD were originally developed as extensions of parametric curves. A parametric curve $g(t)$ is typically written as a linear combination of scalar valued univariate *basis functions* :

$$g(t) = \sum_i \mathbf{g}_i G_i(t),$$

where the $G_i(t)$ are the basis functions associated with a vector valued coefficient \mathbf{g}_i ; the index i ranges over the basis functions for a given curve. If $G_i(u)$ and $G_j(v)$ are sets of univariate basis functions in the parameters u and v respectively, a set of scalar valued bivariate *tensor product* basis functions $G_{i,j}(u, v)$ are formed as products of univariate basis function :

$$G_{i,j}(u, v) = G_i(u)G_j(v).$$

A bivariate tensor product surface $g(u, v)$ can thus be written

$$g(u, v) = \sum_i \sum_j \mathbf{g}_{i,j} G_{i,j}(u, v),$$

where $\mathbf{g}_{i,j}$ are vector valued coefficients.

The domain of a parametric curve $g(t)$ is generally restricted to an interval $a \leq t \leq b$. Under a tensor product construction, the domain of a parametric surface is restricted to a

rectangle $a \leq u \leq b, c \leq v \leq d$. A tensor product surface with this restriction is referred to as a *tensor product patch*. Parametric surface patches with triangular, as opposed to rectangular, domains have also seen use in CAGD. However, the mathematical form of such *triangular patches* is not quite as elementary as that of the tensor product form. For this reason triangular patches have not received the same level of attention as rectangular patches.

The particular form of the basis functions characterize a curve or surface patch. The *Bézier* (Bernstein) (Section 2.4) and *B-spline* (Section 3.2) representations are two bases that are important in this work. In both of these representations, the coefficients can be interpreted as positions, or points in space, and are hence referred to as *control points*. With this interpretation, it is possible to develop geometric intuition about how the shape of the curve or surface will behave based solely on the spatial relationships of its control points.

A Bézier curve of degree d is a smooth approximation to $d + 1$ control points, where the first and last control points are interpolated (see Figure 1.1). Since the curve approximates the control points in an intuitive way, a designer has a good idea of the shape of the curve even before the computer calculates and displays it. Similarly, changing the position of a control point will change the curve in an intuitive and predictable way. Thus the coefficients of the mathematical representation provide a convenient user interface.

The behavior is analogous in tensor product Bézier surface patches. A degree r by s tensor product Bézier patch is a smooth approximation to an $r + 1$ by $s + 1$ array of control points where the corner control points are interpolated (see Figure 1.1). Similarly, a triangular Bézier surface patch of degree d is a smooth approximation to a triangular array of control points where, again, the corner control points are interpolated. Just as with Bézier curves, the shapes of tensor product Bézier patches and triangular Bézier patches are intuitively related to their control points.

As the degree of a Bézier curve or surface goes up, more control points are available to control shape. However, the higher the degree, the less influence a single control point

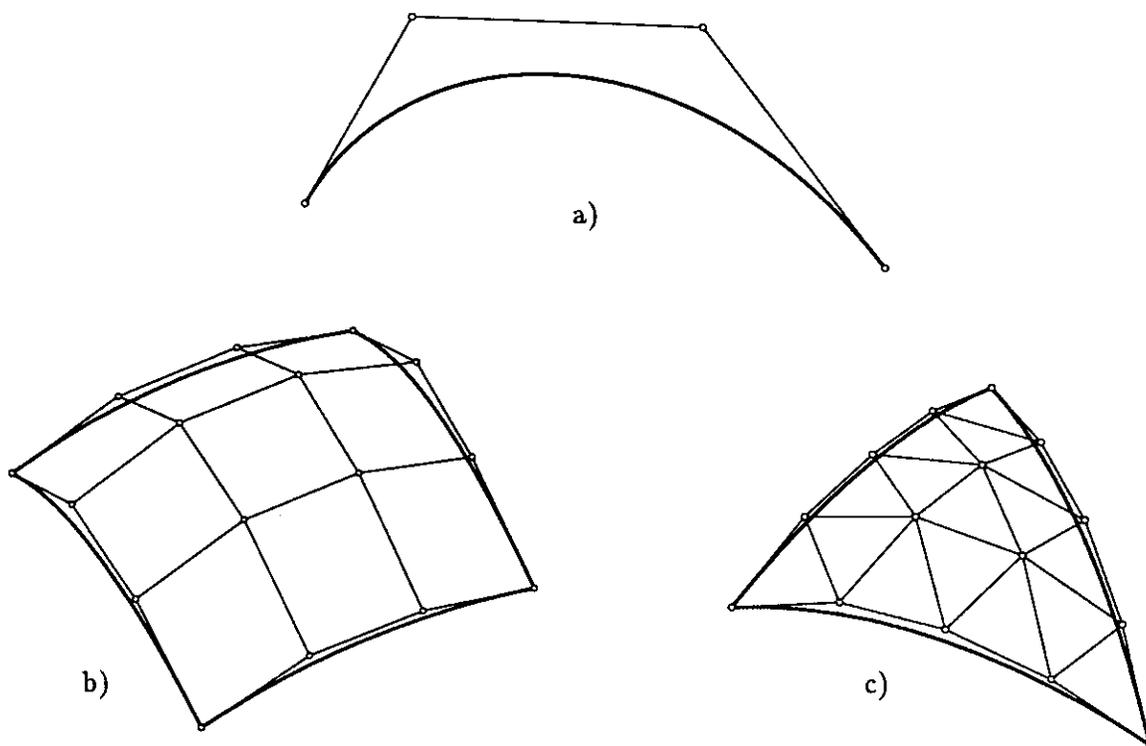


Figure 1.1: Examples of a) a Bézier curve, b) a Bézier tensor product surface, and c) a Bézier triangle

has over this shape. Furthermore, the time required to evaluate the curve or surface increases as a function of the degree. Therefore, it is generally better to construct complex shapes from several individual curve segments or surface patches that fit together smoothly. Unfortunately, the constraints that must be satisfied between the control points of adjacent Bézier elements are often quite complex. Fortunately, other bases exist where the smoothness constraints are more easily satisfied.

B-splines provide such a basis. Unlike the Bézier basis, B-spline basis functions are themselves smooth composite objects. Thus any linear combination of these basis functions will in turn be a smooth composite object. Like Bézier curves and surfaces, a B-spline curve or surface is a smooth approximation to its control points, or *control net* (see Figure 1.2). Similar geometric intuition allows designers to control the shape of a B-spline. In the case of B-splines, the number of control points can increase without

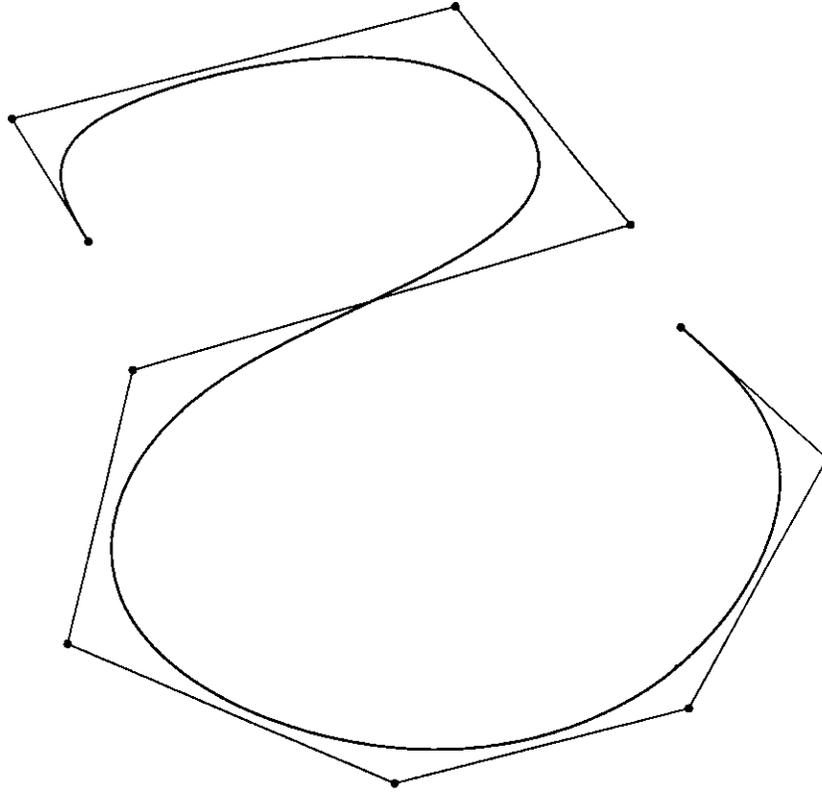


Figure 1.2: A B-spline curve

changing the polynomial degree of the curve or surface. More importantly, B-splines possess a *local control* property. Local control means that if a designer moves a control point the B-spline curve or surface is only affected in a local region near the control point.

Geometrically intuitive behavior based on control points, local control, and automatic satisfaction of the smoothness constraints make B-splines a powerful and popular tool in CAGD. These properties all stem from the mathematical definition of B-splines. Mathematically, a B-spline surface can be thought of as a piecewise deformation of a planar domain, tessellated into a regular grid of rectangles or triangles. The smoothness constraints are analogous to matching various derivatives along patch boundaries. This

notion of smoothness is known as *parametric continuity* (denoted C^k continuity). It is precisely this treatment that limits a B-spline surface to having a strictly planar topological type¹. Next, a quite different approach to surface modeling is considered that does not limit the topological type of a surface a designer wishes to create.

1.2 Implicit Surfaces

Implicit surfaces do not suffer from the topological restrictions found in B-spline surfaces. An implicit surface is defined as the zero set of a trivariate function :

$$f(x, y, z) = 0,$$

where x, y , and z are taken to be the coordinates of a point in 3-space. The surface is defined to be the set of points that satisfy the above equation. The function f is often polynomial in x, y , and z , resulting in a simple algebraic form. Implicit surfaces are well suited to modeling surfaces of arbitrary topological type; the surface is actually a 3 dimensional contour of the trivariate function f , this contour might be any arbitrary 2-manifold.

It is precisely the topological freedom of implicit surfaces that causes trouble with their use. The geometric behavior of an implicit surface can be difficult to control. This can cause the topological type of an implicit surface to change in often unpredictable ways with only a slight perturbation of its defining parameters. Additional problems are encountered with the appearance of “ghost sheets” - pieces of surface that happen to satisfy the implicit equation yet are not intended by a modeler’s definitions.

While this discussion is intended to focus on topology and design considerations, the question of efficiency cannot be ignored. In state-of-the-art (1992) real-time interactive modeling environments, computing with implicit surfaces is generally far too inefficient to be practical. One of the problems involves algorithms used to generate a polygonal

¹Cylindrical and toroidal topological types are also possible if the planar domain is allowed to be “periodic”.

representation of the surface suitable for computer display. These algorithms rely heavily on numerical techniques for computing an approximation to the 3-dimensional contour. Such algorithms are inherently slow and often plagued with stability problems. Despite these difficulties, implicit surfaces remain an effective geometric design tool, especially in non real-time situations calling for organic shapes whose topological types might change over time. Modeling blobs of interacting liquid material, for example, would be exceedingly difficult using any other technique.

1.3 Subdivision Surfaces

Subdivision surfaces are not restricted in topological type like B-splines, and are relatively easy to generate unlike implicit surfaces. A subdivision surface is defined algorithmically. The surface is found as the result of running a subdivision algorithm on an input control mesh. This control mesh is much like the control net of a B-spline scheme but is not required to have a regular structure. The details and behavior of subdivision algorithms varies considerably. For example, the *fractal surfaces* used by Carpenter [Carpenter et al. 82] and others to generate rough mountainous terrain models are a type of subdivision surface. The subdivision surfaces of interest here are used to generate smooth free form surfaces.

Subdivision surfaces are generated by subdividing the input control mesh to obtain a refined mesh with many more control points. The points of the refined mesh are found by a local averaging of points of the original mesh. If this process is repeated, the result is a smooth surface whose shape mimics that of the input control mesh. The earliest such methods were developed by Doo/Sabin [Doo & Sabin 78] and Catmull/Clark [Catmull & Clark 78]. Figure 1.3 illustrates an example of this process.

These early subdivision surfaces were inspired by the subdivision property of B-splines. The subdivision property maintains that a B-spline basis function of degree r can be represented as a linear combination of degree r B-splines of narrower support (see Section 3.1). The implication of the subdivision property is that any B-spline surface

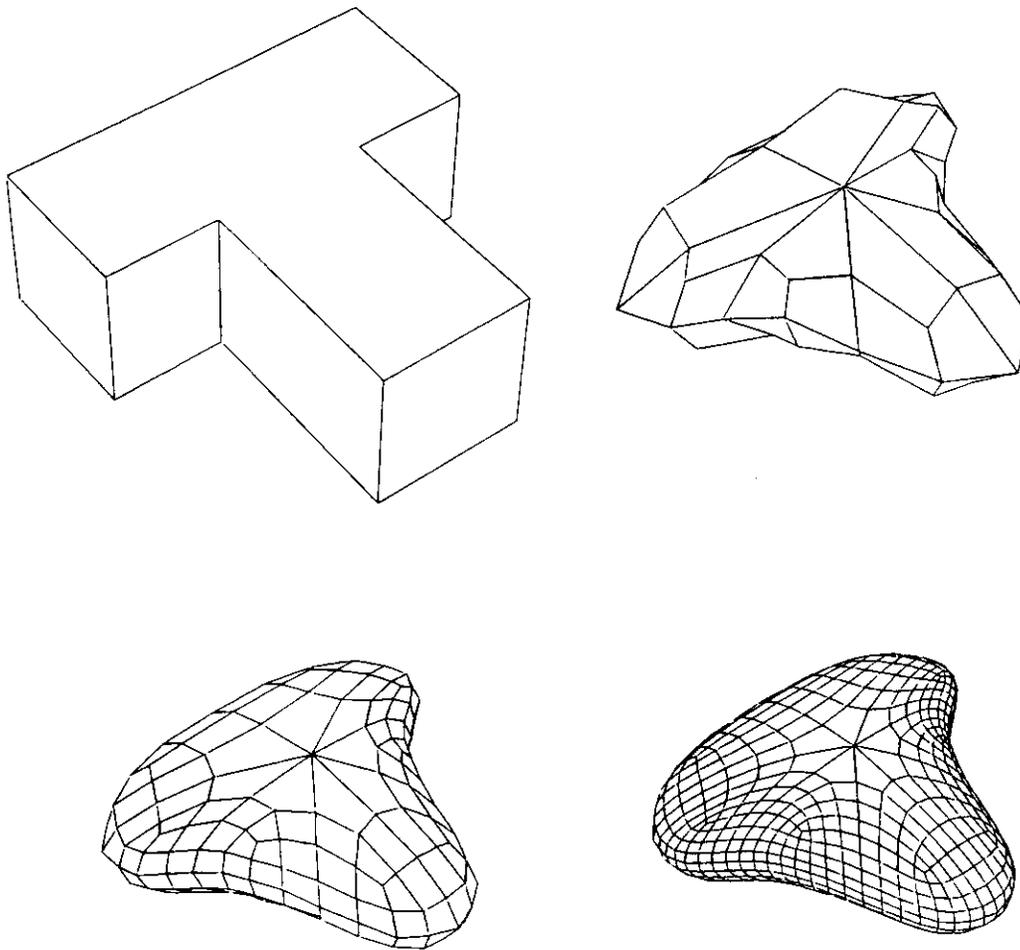


Figure 1.3: A subdivision surface generated by the Catmull & Clark algorithm

with control net \mathcal{A} can be identically represented as a B-spline surface with control net \mathcal{B} , where \mathcal{B} is a refinement of \mathcal{A} . The relationship between \mathcal{A} and \mathcal{B} depends on the degree of the B-splines and the size of the supports of the underlying basis functions. The point is that B-spline surfaces may be considered as a type of subdivision surface. It is known that in the limit, subdivision of a B-spline control mesh will converge to a B-spline surface [Lane & Riesenfeld 80].

Subdivision of a B-spline surface is only defined over regular (rectangular or triangular) control nets. The insight provided by the early subdivision algorithms was to

generalize the refinement algorithms to include irregular control meshes; thus surfaces of arbitrary topological type could be generated. This approach had two advantages. First, since the generalized subdivision schemes are so closely related to regular B-spline subdivision, the surfaces generated by these schemes behave much like regular B-splines. That is, the generalized subdivision surfaces are smooth approximations to their defining control meshes. The second advantage of generalizing B-splines is that these subdivision surfaces *are* B-spline surfaces over locally regular regions of the control mesh. This property is nice because it allows these subdivision techniques to be used in conjunction with B-splines to handle irregularities that might occur in an otherwise regular control mesh.

In general, the only defining characteristic of a subdivision surface is as the limit (when the number of iterations of the algorithm tends to infinity) of a particular algorithm on a particular input control mesh. Subdivision surfaces, in general, cannot be defined analytically (i.e. as a closed form expression). This can lead to practical difficulties when using subdivision surfaces. In addition to evaluation, the various components of a CAGD system might also require derivative (and even curvature) information. Determining this information at arbitrary points on a subdivision surface can be challenging indeed [Doo & Sabin 78]. Alternatively, a subdivision surface can be maintained as a large collection of polygons. There must be sufficiently many polygons in this representation to avoid any polygonal faceting artifacts. In either case, dealing with subdivision surfaces is complex and cumbersome.

1.4 Parametric Surfaces of Arbitrary Topological Type

Subdivision and implicit surfaces allow users to create smooth organic free form shapes that are not restricted in topological type. Neither one, however, has the convenient analytical properties of parametric surfaces nor are as easy to display. What is needed are parametric surfaces schemes that provide the arbitrary topological types of subdivision and implicit surfaces.

A great deal of work has been done toward this end. Particulars of the various problem statements vary, as do the solutions. The most general framework assumes that a control mesh of vertices is given. The problem is to interpolate the vertices with a collection of parametric surfaces. Other variants of this basic problem assume that the vertices of the mesh are tagged with normals adding additional constraints that must be interpolated.

From this general framework, most schemes proceed by constructing a network of curves and cross boundary tangent vectors (a transversal vector field) that interpolates the vertex data. The method of constructing this information varies from scheme to scheme. Some schemes [Barnhill et al. 73, Nielson 79, Charrot & Gregory 84] assume the curve network (with or without the transversal vector fields) to be the input to the problem. In any event, the problem becomes one of finding a set of parametric surface patches that interpolate a boundary curve network (and possibly transversal vector fields) in a smooth fashion. The solutions to this problem vary greatly; for a detailed description of many of these solutions, see [Mann et al. 92].

There are two difficult aspects to the interpolation problem. The first is to construct a smooth join between a pair of parametric surface patches. Notions of parametric continuity must be generalized to remove the limits of a strictly planar topology and gain the freedom necessary to capture surfaces of arbitrary topology. For this, a more general definition of smoothness, called geometric continuity (denoted G^k) is adopted. In particular G^1 continuity, which subsumes strict C^1 continuity and implies that two patches have a common tangent plane at all points along their shared boundary, is sufficient for a pair of surfaces to be considered “smooth”. Many authors [Farin 82, Chiyokura & Kimura 83, Piper 87] provide constructions for joining a pair of patches together smoothly.

A second more subtle aspect of the interpolation problem that causes a significant amount of difficulty is the *twist compatibility* problem. This problem is an artifact of the relationship between the boundary curves and the transversal vector fields. The

interpolation problem assumes that boundary curve endpoints, and endvectors of the transversal vector fields share common positions and tangent planes at mesh vertices. A surface that interpolates this data will be (at least locally) in the continuity class G^1 . Suppose $F_1(u), V_1(u)$ and $F_2(v), V_2(v)$ are a pair of curves and transversal vector fields that share a common endpoint $F_1(0) = F_2(0)$. For simplicity, assume that directions u and v are such that $\frac{dF_1}{du}(0) = V_2(0)$ and $\frac{dF_2}{dv}(0) = V_1(0)$ (if this not the case, cross boundary directions can be chosen without loss of generality such that it is the case). Let $G(u, v)$ be a parametric surface patch that interpolates F_1, V_1, F_2 and V_2 . Interpolating the given data does not guarantee equivalence of the mixed partial derivatives

$$\frac{\partial^2 G}{\partial u \partial v}(0, 0) = \frac{\partial^2 G}{\partial v \partial u}(0, 0).$$

In other words, the constraints on the input are not sufficient to guarantee that $G \in C^2$. If $G \notin C^2$, then G cannot be polynomial, since all polynomials are in the class C^∞ . This dilemma is the central difficulty encountered when constructing a smooth patch network of arbitrary topological type.

Constraining the transversal boundary vectors to be twist compatible is often not possible [Watkins 88, Peters 91]. Therefore, it is impossible to construct an interpolant that fills each region, or face, of a general curve network with a single polynomial surface patch. Solutions to the problem often involve either filling a region with several polynomial patches [Farin 83, Piper 87, Jensen 87, Shirman & Séquin 88], or using rational patches whose mixed partial derivatives may not match (i.e. rational surface that are not of class C^2)[Gregory 74, Chiyokura & Kimura 83].

Solving the problem using a single patch per face can be done for a sufficiently restricted class of meshes [Peters 91]. The restrictions require that only an odd number (or 4) of curves meet at a single vertex. Such curve networks generally only bound at most 3 or 4 sided regions. In this vein, schemes have been developed using exclusively tensor product surfaces that mimic the behavior of B-splines [van Wijk 86, Goodman 91, Lee & Majid 91]. While these techniques are general enough to model surfaces of arbitrary topological type, the limits placed on the structure of the control meshes are

unnatural from a designers point of view.

1.5 Generalized B-splines

The input to the scheme propose here is an irregular, or unrestricted, control mesh; no normals or additional information is needed. Rather than interpolating the vertices of this control mesh, the proposed scheme will approximate the irregular control mesh in much the same way as a B-spline surface approximates a regular control mesh. This behavior is very similar to that of the subdivision surfaces generated by the Catmull/Clark algorithm. Unlike subdivision surfaces, the surfaces generated here are always a smooth (G^1) composite of parametric surface patches. Unlike previous parametric surface solutions, no domain splitting is required and all surface patches are in the class C^∞ . In addition, each surface patch may have any number of boundary curves.

The scheme developed in this thesis approximates, rather than interpolates, the vertices of an irregular control mesh. While the mathematics of this scheme are sufficiently general to support an interpolating method, an approximating method was considered preferable for two reasons. First, approximating methods tend to generate surfaces that have an overall smoother looking shape. Even though an approximating scheme and an interpolating scheme may both produce surfaces that satisfy some mathematical criteria of smoothness, empirical evidence suggests that the surface generated by an approximating scheme generally will appear smoother and be more satisfactory from an aesthetic point of view. Intuitively speaking, one reason for this smoother appearance is that approximation tends to dampen high frequency components better than interpolation. Interpolation is often more critically important in scientific applications where one must be true to some real-world data. In design applications, aesthetic considerations like “looks better” are more important than interpolating a data set that has no real-world counterpart.

The other reason for developing an approximating scheme is to emulate the behavior of B-splines. This was also a feature of the approximating subdivision schemes. Like

those subdivision schemes, the scheme proposed here also generalizes B-spline surfaces. That is, when the control mesh has a regular rectangular structure, the surface is exactly a bicubic tensor product B-spline (see Section 3.2). Furthermore, when the control mesh has a regular triangular structure, the surface is exactly a quartic triangular B-spline (see Section 3.3). These claims are proved in Chapter 6. This relationship to regular B-spline surfaces gives the more general surfaces developed here many of the same desirable design characteristics. There is an intuitive relationship between the control mesh and the resulting surface; moving a control vertex affects the surface in a predictable and local way. In addition, this new surface scheme is compatible with existing B-spline techniques in that it can be used to extend the capabilities of these well known and widely used surface design techniques.

Like many G^1 surface schemes, the scheme developed in this thesis proceeds by constructing a curve network based on the input control mesh. However, unlike many of these schemes that use parametric surface patches, the new scheme does not require any sort of split domain. That is, each hole bounded by the curve network is filled with a single C^∞ parametric surface patch. Furthermore, no limit is placed on the number n of boundary curves possessed by such patches. This construction has a certain mathematical elegance and is more like B-splines than split domain schemes. Additionally, no extraneous boundary curves are introduced. The n -sided patch approach may thereby generate surfaces that are smoother than an approach that requires several sub-patches joined in a G^1 fashion.

The n -sided element used here is the S-patch developed in [Loop & DeRose 89]. S-patches generalize both Bézier tensor product and Bézier triangular patches, thus enabling the proposed scheme to generalize bicubic tensor product and quartic triangular B-splines. The unifying nature of S-patches suggest that a parametric surface modeler can be based on a single patch type. Since all patches are instances of a single general structure, many algorithms may be derived that are independent of the number of sides n , leading to uniformity and simplicity. This contradicts the notion that including n -

sided patches into a geometric modeler increases complexity due to an increase in special cases. While the S-patch representation is in itself more complicated than either tensor product or triangular forms, it has a very nice mathematical structure. This increase in generality arguably offsets the increase in complexity.

1.6 Overview

This thesis is organized as follows. In Chapter 2 a review of Bézier forms is presented including: Bézier simplicies, Bézier curves, Bézier tensor product surfaces and Bézier triangles. Chapter 3 covers the fundamentals of B-splines including: B-spline curves, tensor product B-spline surfaces, and triangular B-splines. Chapter 4 contains the definition and basic properties of S-patches that are necessary in deriving the generalized B-spline scheme. Next, in Chapter 5 the details of the generalized B-spline scheme are presented. In Chapter 6, special cases that arise when restrictions are placed on the connectivity of the control mesh are considered. Finally in Chapter 7, conclusions and directions for future work are given.

Chapter 2

Bézier Forms

A review of the basic mathematics of Bézier forms is now given. This review begins with a few preliminary definitions. Next, Bézier forms of arbitrary degree in any number of variables are defined; these are called *Bézier simplicies* [de Boor 87]. From this general framework, the special cases of *Bézier curves*, *tensor product Bézier surfaces*, and *triangular Bézier surfaces* are considered.

2.1 Preliminary Definitions

An *affine space* X is collection of *points* representing positions, and *vectors* representing directions, together with the following algebra [DeRose 89]:

vector + vector \mapsto vector

scalar \cdot vector \mapsto vector

point - point \mapsto vector

point + vector \mapsto point

The vectors in an affine space shall be denoted by dotted symbols such as \dot{v} , to distinguish them from affine points. A special vector $\dot{0}$, called the *zero vector*, exists such that

$p - p = \vec{0}$, for all points $p \in X$. Although scalar multiplication is not defined for points, a special form called an *affine combination* can be defined within affine geometry. An affine combination is the sum of products of scalars and points as in

$$p = \alpha_1 v_1 + \cdots + \alpha_r v_r,$$

where v_1, \dots, v_r , and p are points, and $\alpha_1, \dots, \alpha_r$ are scalars that sum to 1. A similar *vector combination*, where scalars $\alpha_1, \dots, \alpha_r$ sum to 0, results in an affine vector.

A *k-simplex* Δ is a collection of points v_1, \dots, v_{k+1} such that none of the points can be written as an affine combination of the others. For example, a 1-simplex is a line segment, a 2-simplex is a triangle, a 3-simplex is a tetrahedron and so on. Let u be a point in an affine space X , and let $v_1, \dots, v_{k+1} \in X$ form a k -simplex Δ . If every point $u \in X$ can be written uniquely as an affine combination

$$u = u_1 v_1 + \cdots + u_{k+1} v_{k+1},$$

then X is said to have *dimension k*. The scalars u_1, \dots, u_{k+1} are called the *barycentric coordinates* of u with respect to the simplex Δ .

2.2 Bézier Simplices

A *multi-index* $\vec{i} = (i_1, \dots, i_{k+1})$ is a tuple of non-negative integers whose norm, denoted by $|\vec{i}|$, is defined to be the sum of the components of \vec{i} . Addition, subtraction and scalar multiplication of multi-indices is defined componentwise. A special multi-index \vec{e}_j is defined whose only non-zero component is a one in the j^{th} position (note that $|\vec{e}_j| = 1$).

Let X_1 be an affine space of dimension k and let X_2 be an affine space of arbitrary dimension. Let $\Delta = \{v_1, \dots, v_{k+1}\}$ be a k -simplex in X_1 , and let u_1, \dots, u_{k+1} be the barycentric coordinates of point $u \in X_1$ with respect to Δ . A degree d *Bézier simplex* $Q : X_1 \rightarrow X_2$ is defined by

$$Q(u) = \sum_{|\vec{i}|=d} \mathbf{b}_{\vec{i}} B_{\vec{i}}^d(u_1, \dots, u_{k+1}), \quad (2.1)$$

where the coefficients $\mathbf{b}_{\vec{i}} \in X_2$ are referred to individually as *Bézier control points* and collectively as the *Bézier control net* of Q with respect to Δ . When X_2 is an affine space of dimension one, $Q(u)$ is a scalar valued function and the coefficients $\mathbf{b}_{\vec{i}}$ are referred to as *Bézier ordinates* to distinguish this special case.

The functions $B_{\vec{i}}^d(u_1, \dots, u_{k+1})$ are *Bernstein polynomials* of degree d defined by

$$B_{\vec{i}}^d(u_1, \dots, u_{k+1}) = \binom{d}{\vec{i}} u_1^{i_1} u_2^{i_2} \cdots u_{k+1}^{i_{k+1}},$$

where $\binom{d}{\vec{i}}$ is the multinomial coefficient defined by

$$\binom{d}{\vec{i}} = \frac{d!}{i_1! i_2! \cdots i_{k+1}!}.$$

Bernstein polynomials form a partion of unity, i.e.

$$\sum_{|\vec{i}|=d} B_{\vec{i}}^d(u_1, \dots, u_{k+1}) = 1,$$

and are non-negative, i.e

$$B_{\vec{i}}^d(u_1, \dots, u_{k+1}) \geq 0, \quad 0 \leq u_1, \dots, u_{k+1} \leq 1.$$

These properties imply that Bézier simplices are affine invariant and that $Q(u)$ lies within the convex hull of the control net $\mathbf{b}_{\vec{i}}$ when u lies within the convex hull of Δ . Furthermore, Q approximates its control net and interpolates the corner control points $\mathbf{b}_{d\vec{e}_j}, j = 1, \dots, k + 1$.

2.2.1 Polar Forms

A close relationship between Bézier simplicies and symmetric multiaffine maps has been found by Ramshaw [de Casteljau 63, de Casteljau 86, Ramshaw 87, Ramshaw 88, Ramshaw 89]. A map $q(u_1, \dots, u_d)$ is said to be multiaffine if it is affine when all but one of its arguments are held fixed; it is said to be symmetric if its value does not depend on the ordering of the arguments. Associated with every polynomial $Q : X_1 \rightarrow X_2$ of degree d there is a unique, symmetric d -affine map that agrees with Q on its diagonal

(the diagonal of a function $q(u_1, \dots, u_d)$ is obtained when all arguments are held equal: $Q(u) = q(u, \dots, u)$). This multiaffine map is referred to as the *polarization* or *polar form* of Q (the term *blossom* has also been used to refer to this map).

Ramshaw shows that the Bézier control net of Q relative to the simplex Δ is given by

$$\mathbf{b}_i^r = q(\underbrace{v_1, \dots, v_1}_{i_1}, \underbrace{v_2, \dots, v_2}_{i_2}, \dots, \underbrace{v_{k+1}, \dots, v_{k+1}}_{i_{k+1}}).$$

This insight greatly simplifies the derivation of many procedures involving Bézier simplicies. For example, given the polar form q of a Bézier simplex Q , the control net with respect to another simplex Γ of dimension m is easily found by evaluating q at the vertices of the Γ . The dimension m of Γ need not be the same as the dimension k of Δ . The Bézier simplex corresponding to the simplex Γ will be of dimension m . This shows how each edge of a domain simplex Δ may be treated as a Bézier 1-simplex (a Bézier curve), each face of Δ may be treated as Bézier 2-simplex (a Bézier triangle), and so on.

2.2.2 DeCasteljau's Algorithm

If the Bézier control net of a Bézier simplex is given, its polar form may be evaluated algorithmically. Let $u_1, \dots, u_d \in X_1$, be such that each u_j has barycentric coordinates $u_{j,1}, \dots, u_{j,k+1}$. If q denotes the polar form of Q , the value $q(u_1, \dots, u_d)$ can be computed as follows:

for $r = 1$ to d

for $|\vec{i}| = d - r$

$$\mathbf{b}_i^r \leftarrow u_{r,1} \mathbf{b}_{i+\vec{e}_1}^{r-1} + \dots + u_{r,k+1} \mathbf{b}_{i+\vec{e}_{k+1}}^{r-1}$$

end for

end for

where $\mathbf{b}_i^0 = \mathbf{b}_{\vec{i}}$ and $\mathbf{b}_{0,\dots,0}^d$ is the value $q(u_1, \dots, u_d)$. This procedure is known as *deCasteljau's algorithm*¹.

2.2.3 Derivatives of Bézier Simplices

The derivatives of a Bézier simplex Q are easily expressed in terms of its polar form. Let $\dot{w}_1, \dots, \dot{w}_r$ be r vectors in X_1 . The r^{th} derivative of $Q(u)$ with respect to $\dot{w}_1, \dots, \dot{w}_r$ is written

$$\mathbf{D}_{\dot{w}_1, \dots, \dot{w}_r}^r Q(u) = \frac{d!}{(d-r)!} q(\dot{w}_1, \dots, \dot{w}_r, u, \dots, u). \quad (2.2)$$

Although the barycentric coordinates of a vector (being the difference of two points) sum to zero, deCasteljau's algorithm can be used to evaluate the derivative of a Bézier simplex.

2.2.4 Degree Elevation

A degree d Bézier simplex with control points $\mathbf{b}_{\vec{j}}$ may be represented as an equivalent degree r ($r > d$) Bézier simplex by the following :

$$\mathbf{c}_{\vec{i}} = \sum_{\vec{j}+\vec{k}=\vec{i}} \frac{\binom{d}{\vec{j}} \binom{r-d}{\vec{k}}}{\binom{r}{\vec{i}}} \mathbf{b}_{\vec{j}},$$

where $\mathbf{c}_{\vec{i}}$ are the control points of the degree elevated Bézier simplex.

2.3 Bézier Curves

The Bézier curve form was developed independently by P. Bézier and P. deCasteljau in the late 1950s and early 1960s as a computer aided geometric design tool for use in the French auto industry. A Bézier curve smoothly approximates a set of control points (collectively known as the *control polygon* in the curve case) and interpolates the first and last control points. Figure 2.1 shows an example.

¹Strictly speaking, deCasteljau's algorithm requires that $u_1 = \dots = u_{k+1}$. For lack of a better name, this more general algorithm shall be associated with deCasteljau's name.

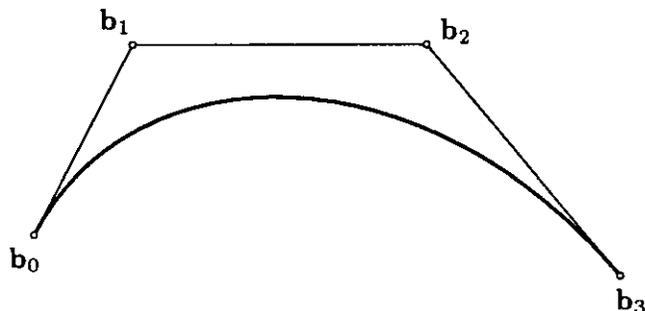


Figure 2.1: A cubic Bézier curve.

Mathematically, a Bézier curve is a special case of a Bézier simplex. If Δ is a 1-simplex, i.e. a line segment, the resulting Bézier form is a curve. The multi-index notation used for Bézier simplices is often too cumbersome when dealing with Bézier curves. More commonly a degree d Bézier curve is written

$$Q(u) = \sum_{i=0}^d \mathbf{b}_i B_i^d(u), \quad 0 \leq u \leq 1,$$

where the coefficients $\mathbf{b}_0, \dots, \mathbf{b}_d$ are the control points and

$$B_i^d(u) = \binom{d}{i} (1-u)^{d-i} u^i,$$

are univariate Bernstein polynomials. It is generally assumed that the domain X_1 of a Bézier curve is the real line and that $u \in [0, 1]$.

2.3.1 The Derivative of a Bézier Curve

The special case of the Bézier simplex derivative formula for the first derivative of a degree d Bézier curve is

$$\frac{d}{du} Q(u) = d \sum_{i=0}^{d-1} (\mathbf{b}_{i+1} - \mathbf{b}_i) B_i^{d-1}(u).$$

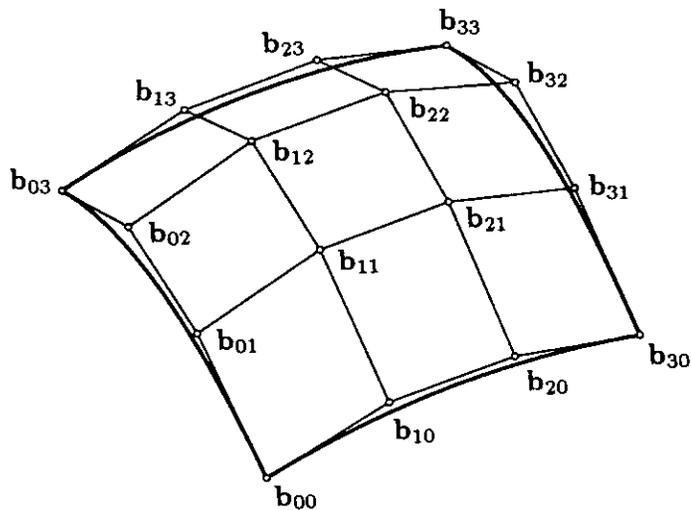


Figure 2.2: A bicubic tensor product Bézier patch

2.4 Tensor Product Bézier Surfaces

While both Bézier and deCasteljau discovered Bézier curves, their extensions to the bivariate setting differed. Bézier based his surface on a tensor product of Bézier curves. A degree r by s tensor product Bézier surface is defined by

$$Q(u, v) = \sum_{i=0}^r \sum_{j=0}^s \mathbf{b}_{i,j} B_i^r(u) B_j^s(v),$$

where the coefficients $\mathbf{b}_{i,j}$, $0 \leq i \leq r$, $0 \leq j \leq s$ are control points that form a rectangular array. An example of a Bézier tensor product patch is shown in Figure 2.2. The domain of a tensor product Bézier surface is generally taken to be the unit square $[0, 1] \times [0, 1]$. A tensor product Bézier surface with a rectangular domain is known as a Bézier tensor product *patch*.

Tensor product Bézier patches have many of the same properties as Bézier simplices such as affine invariance and the convex hull property. These follow from the analogous properties for Bézier curves, and the curve based definition of Bézier tensor product surfaces. Each edge of a tensor product Bézier patch is a Bézier boundary curve. For

example, the image of the edge corresponding to $v = 0$, $u \in [0, 1]$ is the Bézier curve with control points $\mathbf{b}_{0,0}, \dots, \mathbf{b}_{r,0}$.

2.4.1 Derivatives of Tensor Product Bézier Surfaces

A pair of derivative formulae for tensor product Bézier surfaces will be utilized in Chapters 5 and 6. These involve partial derivatives on edges of the rectangular domain. Consider the edge $v = 0$, the partial derivatives of Bézier tensor product patch $Q(u, v)$ are:

$$\mathbf{D}_{\dot{u}} Q(u, 0) = d \sum_{i=0}^{d-1} (\mathbf{b}_{i+1,0} - \mathbf{b}_{i,0}) B_i^{d-1}(u), \quad (2.3)$$

$$\mathbf{D}_{\dot{v}} Q(u, 0) = d \sum_{i=0}^d (\mathbf{b}_{i,1} - \mathbf{b}_{i,0}) B_i^d(u), \quad (2.4)$$

where \dot{u} and \dot{v} correspond to the u and v axes of the domain rectangle.

2.5 Bézier Triangles

The extension of Bézier curves to the bivariate setting by deCasteljau, now called *Bézier triangles*, uses a 3-sided triangular domain. deCasteljau's work lead to the more general theory of Bézier simplicies. In fact, a Bézier triangle is a Bézier simplex where the domain simplex Δ is a triangle (a 2-simplex). The definition of a Bézier triangle is equivalent to that of a Bézier simplex where $k = 2$. Since there are only 3 elements in the corresponding multi-index, the notation is often expanded with $\vec{i} = (i, j, k)$.

A degree d Bézier triangle can then be defined by

$$Q(u) = \sum_{|i+j+k|=d} \mathbf{b}_{i,j,k} B_{i,j,k}^d(u_1, u_2, u_3),$$

where u_1, u_2, u_3 are the barycentric coordinates of u with respect to a triangle Δ , $\mathbf{b}_{i,j,k}$ form a triangular array of Bézier control points, and

$$B_{i,j,k}^d(u_1, u_2, u_3) = \binom{d}{i, j, k} u_1^i u_2^j u_3^k,$$

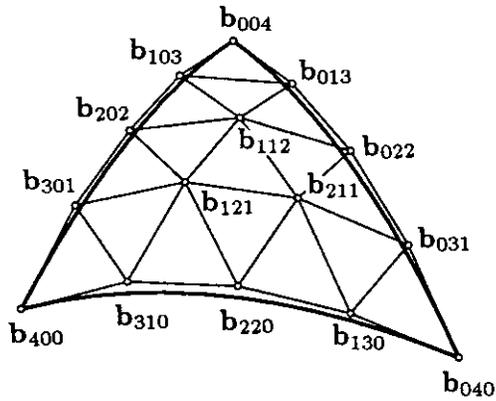


Figure 2.3: A quartic Bézier triangle

are trivariate Bernstein polynomials of degree d . An example of a Bézier triangle is shown in Figure 2.3. Properties such as convex hull and affine invariance follow directly from those of Bézier simplicies.

Chapter 3

B-splines

In this chapter, B-spline curves, tensor product B-spline surfaces, and triangular B-splines will be discussed. Basic definitions and properties are given. Particular attention is paid to relationships between B-spline and Bézier forms.

3.1 B-spline Curves

Intuitively, a B-spline curve is a smooth approximation to a set of control points. The control points are collectively referred to as the *de Boor polygon*. Unlike a Bézier curve, a B-spline curve is piecewise polynomial, i.e. a composite of one or more polynomial curve segments. While it is possible to design smooth composite objects with Bézier curves, satisfying the continuity constraints imposed on the positions of the Bézier control points of adjacent curve segments can be quite complicated. B-splines curves are free from these difficulties, automatically satisfying the continuity constraints of adjacent polynomial curve segments.

A degree r , piecewise polynomial, B-spline curve $S^r(u)$ is defined by

$$S^r(u) = \sum_i \mathbf{d}_i N_i^r(u). \quad (3.1)$$

The collection of point valued coefficients \mathbf{d}_i are often referred to as the *de Boor points* or *de Boor polygon*. The $N_i^r(u)$ are normalized B-spline basis functions of degree r defined

over a sequence of knots $\{i\}$. This knot sequence, or *knot vector* forms a partition of the real u -axis. Only the case of equidistant knot spacing is important here. For simplicity, knot sequences will be derived from the sequence

$$\mathcal{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}.$$

Under these restrictions, (3.1) becomes

$$S^r(u) = \sum_{i \in \mathcal{Z}} \mathbf{d}_i N^r(u - i). \quad (3.2)$$

The $N^r(u - i)$ are translates of a single B-spline $N^r(u)$.

The $N^r(u)$ form a partition of unity, i.e.

$$\sum_{i \in \mathcal{Z}} N^r(u - i) = 1.$$

Hence B-spline curves are affine invariant. B-spline basis functions are non-negative, i.e.

$$N^r(u) \geq 0.$$

This condition implies that a B-spline curve lies in the convex hull of its de Boor points. Each $N^r(u)$ has local support, that is

$$N^r(u) = 0 \quad \text{if } u \notin [0, r + 1].$$

Thus moving a single de Boor point only effects a B-spline curve locally. $N^r(u)$ is $(r - 1)$ times continuously differentiable. Therefore, a B-spline curve $S^r \in C^{r-1}$.

B-spline basis functions may be defined in several different ways[Dæhlen & Lyche 91].

Most commonly they are defined recursively:

$$N^r(u - i) = \frac{u - i}{r} N^{r-1}(u - i) + \frac{i + r + 1 - u}{r} N^{r-1}(u - i - 1),$$

where

$$N^0(u) = \begin{cases} 0, & \text{if } u \in [0, 1] \\ 1, & \text{otherwise.} \end{cases}$$

B-spline basis functions can also be defined via a convolution formula:

$$N^{r+1}(t) = \int_{-\infty}^{\infty} N^0(t-u)N^r(u)du.$$

This convolution approach is the key to defining triangular B-splines (see Section 3.3).

A B-spline curve may also be *subdivided*. That is, one can write

$$S^r(u) = \sum_{j \in \mathcal{Z}/2} \hat{\mathbf{d}}_j N^r(2(u-j)),$$

where $\hat{\mathbf{d}}_j$ is a refined de Boor polygon and $N^r(2(u-j))$ are B-spline basis functions with half the support of $N^r(u)$. The relationship between \mathbf{d}_i and $\hat{\mathbf{d}}_j$ is straightforward [Lane & Riesenfeld 80], but not important here.

Rather than evaluating the various basis functions and taking the sum of products with the de Boor points, a B-spline curve may be evaluated directly from the de Boor points using the *de Boor algorithm*:

$$\mathbf{d}_i^k(u) = \frac{i+r-k-u}{r-k+1} \mathbf{d}_{i-1}^{k-1}(u) + \frac{u-i+1}{r-k+1} \mathbf{d}_i^{k-1}(u), \quad k = 1, \dots, r,$$

where

$$\mathbf{d}_i^0(u) = \mathbf{d}_i.$$

The point $\mathbf{d}_i^r(u)$ is the point on the curve corresponding to $S^r(u)$.

3.1.1 B-spline Curves in Bézier Form

Since B-splines and Bézier forms are both representations of polynomials, it follows that algorithms to convert between these two representations exist. Of particular interest here is the conversion from a degree r B-spline curve into collections of degree r Bézier curve segments. Algebraically, this conversion can be viewed as a simple change of polynomial basis, from a B-spline basis to the Bézier, or Bernstein, basis. Geometrically, the conversion is viewed as constructing the Bézier control points of individual curve segments by taking affine combinations of the de Boor points. The particular affine combinations needed to convert a B-spline representation into a Bézier representation

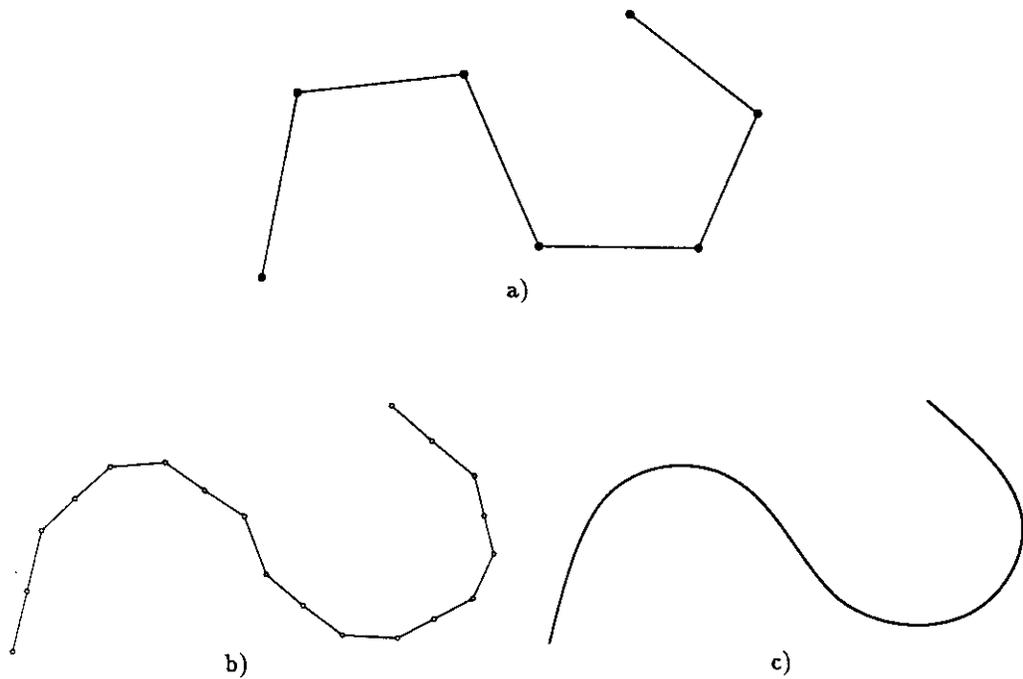


Figure 3.1: a) A de Boor polygon b) The corresponding Bézier control points c) The resulting curve

can be found by solving various linear systems. More direct geometric constructive approaches have been found by Boehm[Boehm 77, Boehm 80, Boehm 81, Boehm 82b] and Sablonnière[Sablonnière 78]. Figure 3.1 shows an example of a de Boor polygon, the corresponding Bézier control points, and the resulting curve.

For the case of uniform knot spacing, the affine combinations needed to convert from a B-spline representation to a Bézier representation are limited to only a few simple cases for each degree r curve. The following are the cases $r = 1, 2, 3$ where \mathbf{b}_j denotes the j^{th} Bézier control point of the i^{th} curve segment:

Linear case, $r = 1$

$$\mathbf{b}_0 = \mathbf{d}_i,$$

$$\mathbf{b}_1 = \mathbf{d}_{i+1}.$$

Quadratic case, $r = 2$

$$\begin{aligned}\mathbf{b}_0 &= \frac{1}{2}\mathbf{d}_{i-1} + \frac{1}{2}\mathbf{d}_i, \\ \mathbf{b}_1 &= \mathbf{d}_i, \\ \mathbf{b}_2 &= \frac{1}{2}\mathbf{d}_i + \frac{1}{2}\mathbf{d}_{i+1}.\end{aligned}$$

Cubic case, $r = 3$

$$\begin{aligned}\mathbf{b}_0 &= \frac{1}{6}\mathbf{d}_{i-1} + \frac{2}{3}\mathbf{d}_i + \frac{1}{6}\mathbf{d}_{i+1}, \\ \mathbf{b}_1 &= \frac{2}{3}\mathbf{d}_i + \frac{1}{3}\mathbf{d}_{i+1}, \\ \mathbf{b}_2 &= \frac{1}{3}\mathbf{d}_i + \frac{2}{3}\mathbf{d}_{i+1}, \\ \mathbf{b}_3 &= \frac{1}{6}\mathbf{d}_i + \frac{2}{3}\mathbf{d}_{i+1} + \frac{1}{6}\mathbf{d}_{i+2}.\end{aligned}$$

From these relationships it is possible to treat the de Boor points of a B-spline as a kind of *scaffolding* from which a collection of Bézier curve segments meeting with C^{r-1} are constructed. This point of view is fundamental to achieving the results of this thesis.

3.2 Tensor Product B-spline Surfaces

A piecewise polynomial tensor product B-spline surface $S^{r,s}(u, v)$ is defined by

$$S^{r,s}(u, v) = \sum_{i \in \mathcal{Z}} \sum_{j \in \mathcal{Z}} \mathbf{d}_{i,j} N^{r,s}(u - i, v - j), \quad (3.3)$$

where the coefficients $\mathbf{d}_{i,j}$ form a rectangular lattice of control points referred to as the *de Boor net*. $N^{r,s}(u - i, v - j)$ are normalized tensor product B-splines of degree r by s defined over the knot vectors $\{i\}$ and $\{j\}$. The knot set $\{i \times j\}$ forms a rectangular grid of points.

A tensor product B-spline is the product of two independently parameterized univariate B-splines, i.e.,

$$N^{r,s}(u - i, v - j) \equiv N^r(u - i)N^s(v - j),$$

where $N^r(u - i)$ and $N^s(v - j)$ are univariate B-splines of degree r and s respectively. It follows that many of the properties and formulae for tensor product B-spline surfaces can be derived directly from the theory of univariate B-splines. The $N^{r,s}(u - i, v - j)$ form a partition of unity and are non-negative. These conditions imply that a tensor product B-spline surface is affine invariant and lies in the convex hull of the de Boor net. $N^{r,s}(u, v)$ has local support; thus a change in a single de Boor point will only affect the surface locally. $N^{r,s}(u, v)$ is $r - 1$ and $s - 1$ times continuously differentiable in the u and v directions respectively.

It is possible to demonstrate subdivision, recursion, and convolution formulae for tensor product B-spline basis functions; these follow directly from their univariate counterparts. de Boor's algorithm can be used to evaluate a tensor product B-spline surface by holding v constant and treating each row of the de Boor net as a de Boor polygon; each of the corresponding 'curves' is evaluated in u , and the resulting column of values is treated as the de Boor points of a curve in v and evaluated to get a point on the surface.

3.2.1 Tensor Product B-spline Surfaces in Bézier Form

Since the tensor product B-splines $N^{r,s}(u, v)$ are piecewise polynomial, it should be possible to represent a tensor product B-spline surface as a collection of tensor product Bézier patches. As in the curve case, changing a B-spline representation to a Bézier representation is equivalent to changing the polynomial basis. Geometric constructions for this procedure can be derived from those given previously in the curve case. Figure 3.2 shows a rectangular de Boor net, the corresponding tensor product Bézier patches, and the resulting surface.

The bicubic case ($r = s = 3$) is given special attention in this work. In this case the corresponding polynomial patches are the lowest degree patches that are smooth (at least C^1), symmetric ($r = s$), and in one-to-one correspondence with faces of the de Boor net. That is, for each de Boor net face $\{\mathbf{d}_{i,j}, \mathbf{d}_{i+1,j}, \mathbf{d}_{i+1,j+1}, \mathbf{d}_{i,j+1}\}$, there is exactly one bicubic patch (ignoring any boundary conditions). In the bicubic case, given below, the

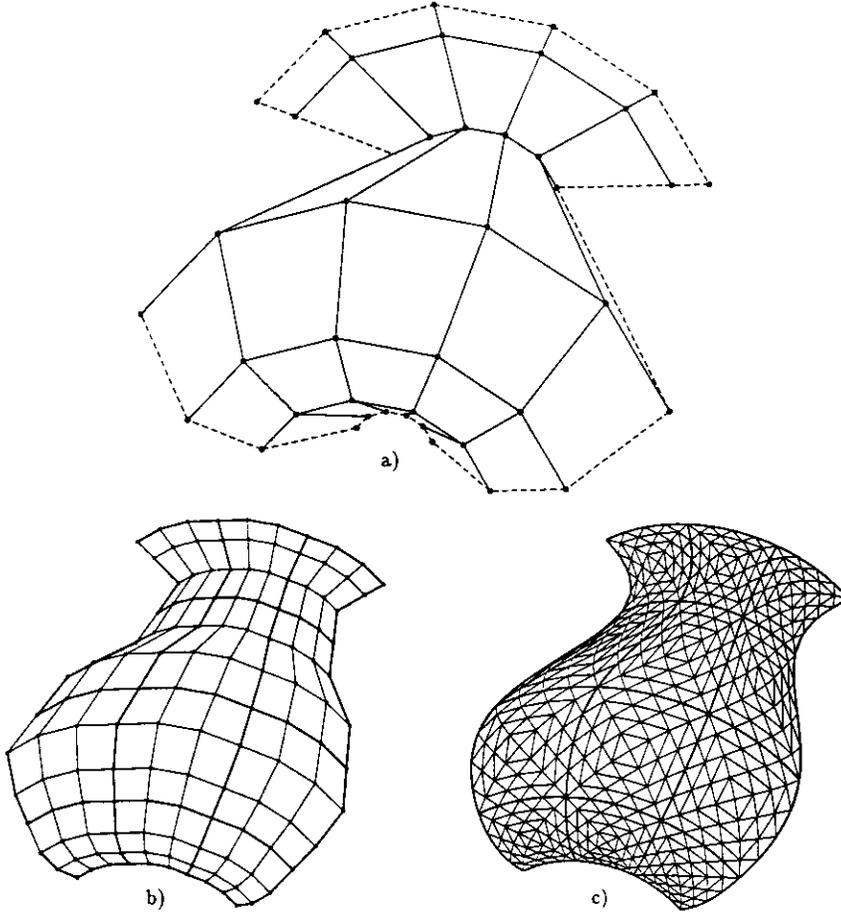


Figure 3.2: a) A rectangular de Boor net b) The corresponding tensor product Bézier control nets c) The resulting surface

points $\mathbf{b}_{k,l}$ are the Bézier controls point of the patch corresponding to such a face:

Bicubic tensor product case, $r = s = 3$

$$\begin{aligned} \mathbf{b}_{00} = & \frac{4}{9}\mathbf{d}_{i,j} + \frac{1}{9}[\mathbf{d}_{i+1,j} + \mathbf{d}_{i,j+1} + \mathbf{d}_{i-1,j} + \mathbf{d}_{i,j-1}] \\ & + \frac{1}{36}[\mathbf{d}_{i+1,j+1} + \mathbf{d}_{i-1,j+1} + \mathbf{d}_{i-1,j-1} + \mathbf{d}_{i+1,j-1}], \end{aligned} \quad (3.4)$$

$$\mathbf{b}_{10} = \frac{4}{9}\mathbf{d}_{i,j} + \frac{2}{9}\mathbf{d}_{i+1,j} + \frac{1}{9}[\mathbf{d}_{i,j+1} + \mathbf{d}_{i,j-1}] + \frac{1}{18}[\mathbf{d}_{i+1,j+1} + \mathbf{d}_{i+1,j-1}], \quad (3.5)$$

$$\mathbf{b}_{11} = \frac{4}{9}\mathbf{d}_{i,j} + \frac{2}{9}[\mathbf{d}_{i+1,j} + \mathbf{d}_{i,j+1}] + \frac{1}{9}\mathbf{d}_{i+1,j+1}. \quad (3.6)$$

The remaining 13 Bézier control points for this patch are found by formulae symmetric

to those above.

3.3 Triangular B-spline Surfaces

In his 1976 Ph.D. thesis, Sabin [Sabin 76] discovered a triangular form of B-spline surfaces. A triangular B-spline is written

$$S^{r,s,t}(u, v) = \sum_{i,j} \mathbf{d}_{i,j} N^{r,s,t}(u - i, v - j),$$

where the $\mathbf{d}_{i,j}$ are interpreted as a triangular de Boor net. A typical face of this triangular de Boor net is labeled $\{\mathbf{d}_{i,j}, \mathbf{d}_{i+1,j}, \mathbf{d}_{i+1,j+1}\}$. That is, the triangular de Boor net is labeled just a rectangular net with diagonal edges $(\mathbf{d}_{i,j}, \mathbf{d}_{i+1,j+1})$.

The $N^{r,s,t}(u, v)$ form a partition of unity, are non-negative, have local support, and are $r - 1$, $s - 1$ and $t - 1$ times continuously differentiable in the r , s and t directions respectively. Triangular B-splines can be generated recursively [Boehm 85b], and may be subdivided [Boehm 83]. A recursive evaluation procedure in the spirit of de Boor's algorithm also exists for triangular B-splines [Boehm 82a].

Sabin originally generated triangular B-spline basis functions by r , s , and t convolutions in each of the three directions of a triangular grid. This definition of triangular B-splines involves triple integrals and is mainly of theoretical interest. Triangular B-splines have also been shown to be a member of a more general class of splines called *box splines* [Dahmen & Micchelli 84, Boehm 85a, Boehm 85b]. In the box spline approach, the basis functions are found as the *shadow* of a hyper-cube in an m -dimensional affine space, projected onto a 2-dimensional affine space. The height of the basis function at any point is proportional to the volume of the hyper-cube that is projected to that point. Again, this definition is primarily a theoretical one.

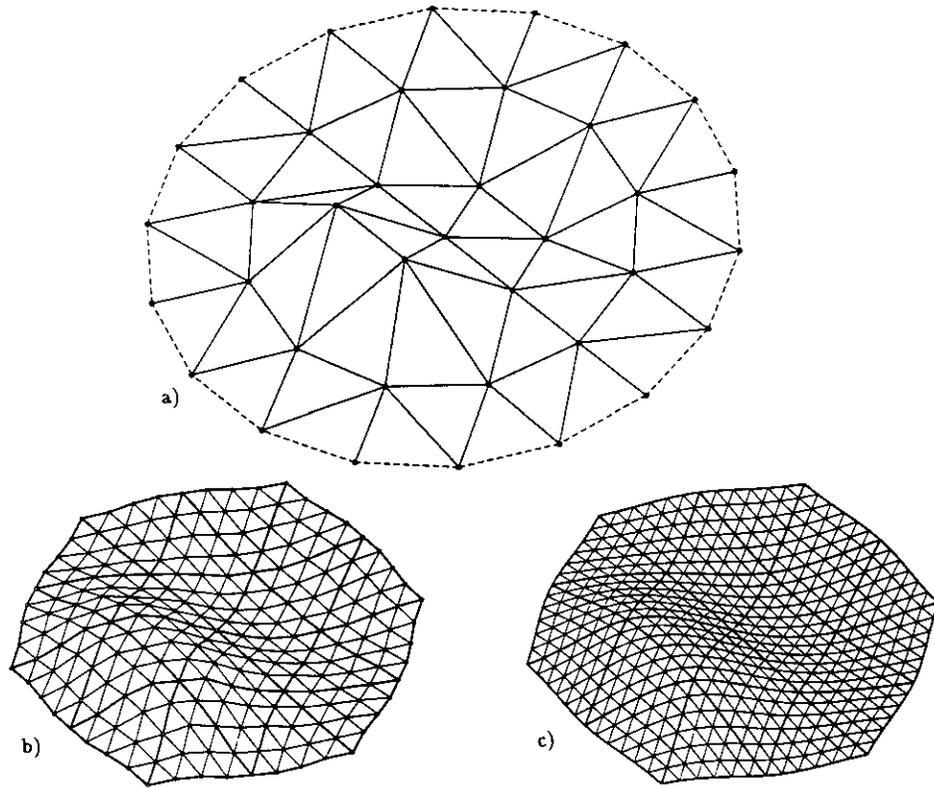


Figure 3.3: a) A triangular de Boor net b) The corresponding triangular Bézier control nets c) The resulting surface

3.3.1 Triangular B-splines in Bézier Form

For practical purposes, Sabin devised an algorithm (based on convolutions) for generating the Bézier ordinates¹ of a triangular B-spline basis function $N^{r,s,t}(u, v)$. Once the Bézier ordinates are known, it is possible to determine affine combinations for constructing the Bézier control points of triangular patches from the triangular de Boor net. An algorithm for computing the Bézier control points of triangular B-spline directly from the de Boor net can be found in [Boehm 82b]. Figure 3.3 shows a triangular de Boor net, the corresponding triangular Bézier patches, and the resulting surface.

¹Bézier ordinates are the control points of scalar valued Bézier forms

The case $r = s = t = 2$, like the tensor product bicubic B-spline case, is given special attention. Again, in this case the resulting triangular patches are the lowest degree, smooth, symmetric ($r = s = t$), patches that are in one-to-one correspondence with the triangular faces of the de Boor net. Since the triangular patches associated with the case $r = s = t = 2$ are degree 4, this case is also known as a *quartic triangular B-spline surface*. Below are the formulae for constructing the Bézier control points $\mathbf{b}_{k,l,m}$ of a quartic triangular Bézier patch corresponding to a de Boor net face:

Quartic triangular B-spline case, $r = s = t = 2$

$$\begin{aligned} \mathbf{b}_{400} &= \frac{1}{2}\mathbf{d}_{i,j} \\ &+ \frac{1}{12}[\mathbf{d}_{i+1,j} + \mathbf{d}_{i+1,j+1} + \mathbf{d}_{i,j+1} + \mathbf{d}_{i-1,j} + \mathbf{d}_{i-1,j-1} + \mathbf{d}_{i,j-1}], \end{aligned} \quad (3.7)$$

$$\begin{aligned} \mathbf{b}_{310} &= \frac{1}{2}\mathbf{d}_{i,j} + \frac{1}{6}\mathbf{d}_{i+1,j} + \frac{1}{8}[\mathbf{d}_{i+1,j+1} + \mathbf{d}_{i,j-1}] \\ &+ \frac{1}{24}[\mathbf{d}_{i,j+1} + \mathbf{d}_{i-1,j-1}], \end{aligned} \quad (3.8)$$

$$\mathbf{b}_{220} = \frac{1}{3}[\mathbf{d}_{i,j} + \mathbf{d}_{i+1,j}] + \frac{1}{6}[\mathbf{d}_{i+1,j+1} + \mathbf{d}_{i,j-1}], \quad (3.9)$$

$$\mathbf{b}_{211} = \frac{5}{12}\mathbf{d}_{i,j} + \frac{1}{4}[\mathbf{d}_{i+1,j} + \mathbf{d}_{i+1,j+1}] + \frac{1}{24}[\mathbf{d}_{i,j+1} + \mathbf{d}_{i,j-1}]. \quad (3.10)$$

The remaining 8 Bézier points for this patch are found by formulae that are symmetric to those just given.

Chapter 4

S-patches

S-patches are a generalization of Bézier surfaces where any number n of boundary curves are permissible. A detailed description of the underlying theory can be found in [Loop & DeRose 89]. In this chapter, only those results necessary in the development of the present work are summarized.

4.1 S-patch Basics

S-patches possess a rich structure, largely because they are defined in terms of multivariate Bernstein polynomials and Bézier simplexes. An n -sided S-patch S is a mapping from a domain n -gon P in a 2-dimensional affine space, to some model space M of arbitrary dimension. S is conceptually constructed in two phases: first, $P = \{p_1, \dots, p_n\}$ is embedded into an intermediate domain simplex $\Delta = \{v_1, \dots, v_n\}$ contained in an affine space Y of dimension $n - 1$; next a Bézier simplex is created using Δ as its domain and M as its range; finally, S is defined as the composition of the embedding and the Bézier simplex. That is, if $L : P \rightarrow \Delta$ represents the embedding, and if $B : \Delta \rightarrow M$ is the Bézier simplex, then

$$S(p) = B \circ L(p), \quad p \in P, \quad (4.1)$$

as indicated in Figure 4.1.

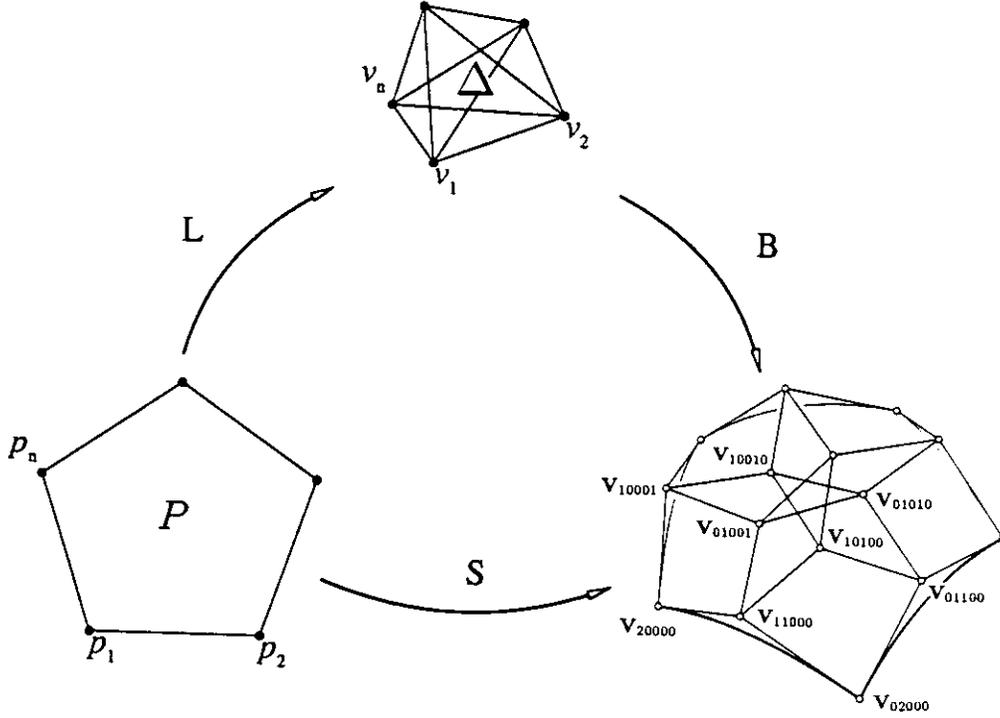


Figure 4.1: Schematic representation of an S-patch.

Several helpful definitions are introduced in order to describe the embedding L used to map the domain polygon P into the intermediate simplex Δ . Let functionals $\alpha_i(p)$ denote the ratio of the signed area of the triangle $pp_i p_{i+1}$ to the area of the triangle $p_i p_{i+1} p_{i+2}$, where the sign is chosen to be positive if p is inside P . (Note: all indices are to be treated in cyclic fashion.) The geometry of the functionals $\alpha_1, \dots, \alpha_n$ is illustrated in Figure 4.2. Let

$$\pi_i(p) = \alpha_1(p) \cdots \alpha_{i-2}(p) \cdot \alpha_{i+1}(p) \cdots \alpha_N(p),$$

for $i = 1, \dots, n$, denote the product of all functionals except for α_{i-1} and α_i , and let

$$\ell_i(p) = \frac{\pi_i(p)}{\pi_1(p) + \cdots + \pi_n(p)}.$$

With these definitions, every point $p \in P$ is mapped by L into the point

$$L(p) = \ell_1(p)v_1 + \ell_2(p)v_2 + \cdots + \ell_n(p)v_n, \quad (4.2)$$

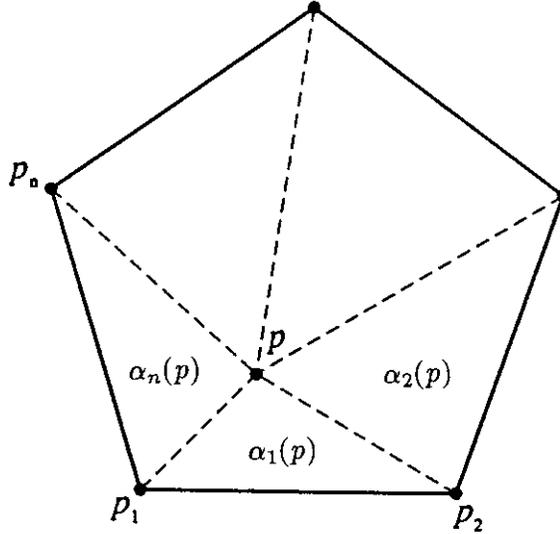


Figure 4.2: The geometry of functionals $\alpha_1(p), \dots, \alpha_n(p)$.

in Y . This embedding has the important property that it is *edge-preserving*, meaning that if p lies on an edge of P , then $L(p)$ lies on an edge of Δ ; additionally, the interior of P is mapped into the interior of Δ [Loop & DeRose 89].

If $\mathbf{b}_{\vec{i}}$ denotes the control net of a Bézier simplex B , then an S-patch S is defined as

$$S(p) = B \circ L(p) = \sum_{\vec{i}} \mathbf{b}_{\vec{i}} B_{\vec{i}}^d(\ell_1(p), \dots, \ell_n(p)). \quad (4.3)$$

The integer d in Equation 4.3 is known as the *depth* of the S-patch, to avoid confusion with the polynomial degree of the patch, which is $d(n-2)$. The control net $\mathbf{b}_{\vec{i}}$ is taken as the control net of S , an example of which is shown in Figure 4.1.

The compositional structure of the S-patch S together with the edge-preserving character of the embedding L endows the S-patch representation with a number of useful properties. For example, S-patches are affine invariant and possess the convex hull property; these properties are inherited from Bézier simplices. DeCasteljau's algorithm can be used to evaluate S-patches, once the component functions $\ell_1(p), \dots, \ell_n(p)$ have been determined. Similarly, given an S-patch control net of depth d describing a patch $S = B \circ L$,

the S-patch control net of depth $d + 1$ for S can be constructed by executing the Bézier simplex degree raising algorithm (Section 2.2.4) on the control net of B . Thus, S-patch control nets can be *depth elevated*.

A very useful property of S-patches is a consequence of the edge preserving nature of the embedding map $L(p)$. Recall that the image of an edge of the domain Δ under a Bézier simplex is a Bézier curve (see Section 2.2.1). Since edges of P are taken to edges of Δ under L , the boundary curves of an S-patch must be Bézier curves. This fact has important practical implications. In order for a pair of S-patches to meet with C^0 continuity, they need only share control points along a common boundary curve. Such control points along a boundary curve are referred to as *boundary points*.

S-patch control nets consist of interconnected n -sided closed polygonal *panels*. For instance, in Figure 4.1 the points \mathbf{V}_{20000} , \mathbf{V}_{11000} , \mathbf{V}_{10100} , \mathbf{V}_{10010} , \mathbf{V}_{10001} form one such panel. The panels of a control net that contain boundary points are termed *boundary panels*. The tangent plane variation along a boundary curve is determined entirely by the corresponding boundary panels.

4.2 Bézier Surfaces are Generalized

One of the most significant properties of S-patches is that they generalize Bézier surfaces. That is, when $n = 3$, the S-patch form reduces to a Bézier triangle, and when $n = 4$, S-patches coincide with Bézier tensor product patches (for proofs see [Loop & DeRose 89]). This property is interesting theoretically and has many practical benefits. Until the development of S-patches, the theories (and implementations) of triangular and tensor product Bézier patches were treated separately. These surfaces were clearly similar in behavior, but each had its own peculiarities. Some (cf. [Farin 90]) debated the merits of using one form over the other, since it was believed that using both triangular and tensor product Bézier patches in a single modeling system would result in an increase in software complexity due to the need for separate treatments of the two distinct types. The unification of these two types of Bézier surfaces into a single patch type (the S-patch)

shows that this is not the case.

In the case $n = 3$, S-patches and Bézier triangles are equivalent. However when $n = 4$, S-patches and by- d -ic tensor product Bézier patches are not quite the same, although they are very closely related. Let $\mathbf{w}_{i,j}$ be the Bézier control points of a bi- d -ic tensor product Bézier patch, and let $\mathbf{v}_{\vec{i}}$ be the control points of a depth d S-patch that represents the same surface. The $\mathbf{w}_{i,j}$ and $\mathbf{v}_{\vec{i}}$ are related by the following two formulae (see [Loop & DeRose 89] for a proof):

$$\mathbf{w}_{i,j} = \sum_{\substack{\vec{i} \\ i_2+i_3=i \\ i_3+i_4=j}} \frac{\binom{d}{\vec{i}}}{\binom{d}{i} \binom{c}{j}} \mathbf{v}_{\vec{i}},$$

$$\mathbf{v}_{\vec{i}} = \mathbf{w}_{i_2+i_3, i_3+i_4},$$

where multi-index $\vec{i} = (i_1, i_2, i_3, i_4)$.

The generalization of Bézier surfaces is important in the thesis for the following reason: since the goal of this work is to create a B-spline like surface scheme that includes both bicubic B-spline and quartic triangular B-spline surfaces as proper subsets, it is absolutely essential that a patch type exists that is general enough to represent the individual surface patches generated by these two schemes. S-patches are ideally suited to this purpose.

4.3 Representing Polynomials

Let X be an affine space of dimension 2 and let M be an affine space of arbitrary dimension. Let $Q : X \rightarrow M$ be a polynomial of degree d having polar form q . The S-patch representation of depth d for Q with respect to the polygon p_1, \dots, p_n has control points given by

$$\mathbf{b}_{\vec{i}} = q(\underbrace{p_1, \dots, p_1}_{i_1}, \underbrace{p_2, \dots, p_2}_{i_2}, \dots, \underbrace{p_n, \dots, p_n}_{i_n}). \quad (4.4)$$

For a proof see Claim 6.3 [Loop & DeRose 89]. The importance of this fact here is that Equation (4.4) provides an algorithm for representing a degree d polynomial as a depth

d S-patch. This principle is important when joining S-patches together smoothly.

In order to appreciate the significance of Equation (4.4), it is important to understand the associated geometry. If all but one argument of q is held fixed, the resulting map is, by definition, affine in the one remaining argument. In each S-patch panel, the polar form associated with each control point of the panel differs in only one argument. Hence each S-patch panel is the image of polygon p_1, \dots, p_n under an affine map. Therefore, in the S-patch control net found by Equation (4.4) all S-patch panels are planar, affine n -gons.

4.4 Joining S-patches Smoothly

In this Section, the precise definition of what is meant by a smooth join between a pair of S-patches is given. The constructive approach used in this thesis is then outlined and shown to satisfy this definition, at least in the absence of parametric irregularities.

“Smooth” in this context refers to first order *geometric continuity*, denoted G^1 . If a collection \mathcal{S} of two or more patches meet with G^1 continuity, then the notation $\mathcal{S} \in G^1$ is used to denote this fact [DeRose 85]. Informally, G^1 means that a pair of surface patches have a continuous tangent plane along a shared boundary. However, this simple notion does not exclude a ‘razor sharp’ join between a pair of patches (i.e. one where the patches lie on the same side of their common boundary), or a join between patches whose boundary tangent planes degenerate. For these reasons, a more precise definition of first order geometric continuity is needed.

Before giving the definition of G^1 continuity, a few preliminary definitions are in order. Let $S : P \rightarrow M$ and $T : Q \rightarrow M$ be a pair of S-patches with domain polygons $P = \{p_1, \dots, p_n\}$ and $Q = \{q_1, \dots, q_m\}$ respectively (see Figure 4.3). Let $\dot{s} = p_2 - p_1$ and $\dot{t} = p_n - p_1$ be a pair of directions in the domain of S , and $\dot{r} = q_n - q_1$ and $\dot{w} = q_2 - q_1$ be a pair of directions in the domain of T . Let $E_p(t) = p_1 + t\dot{s}$ and $E_q(t) = q_1 + t\dot{r}$ be a pair of functions that map $[0, 1]$ into the domains of S and T respectively.

The following definition contains 4 conditions that are generally accepted for a pair

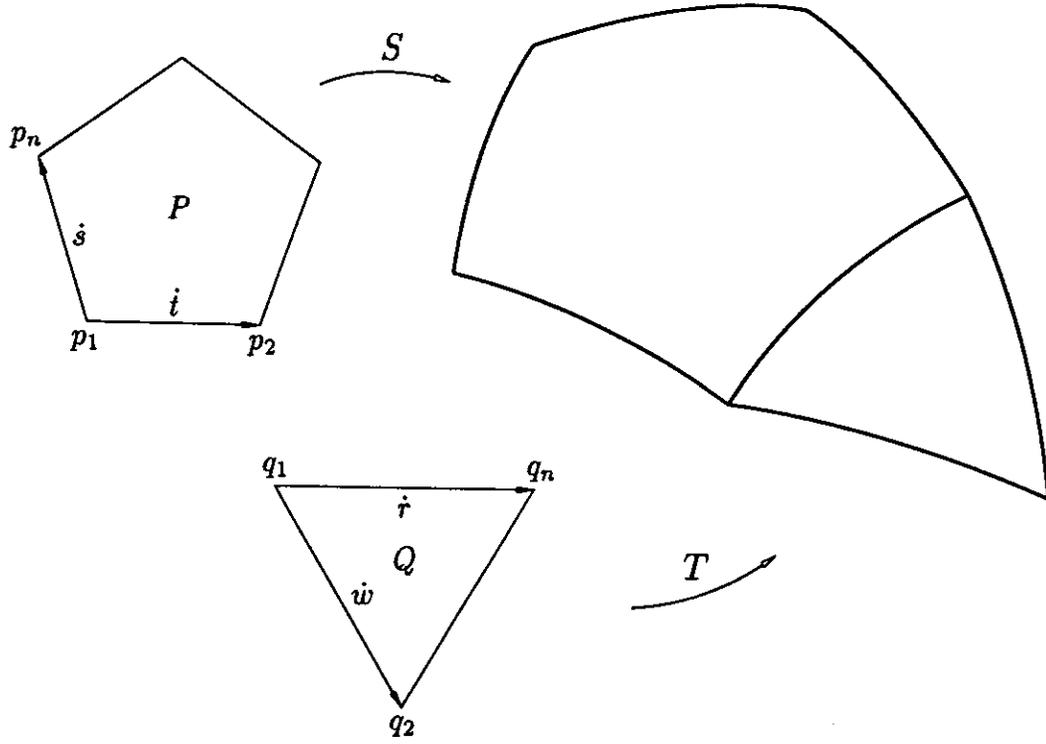


Figure 4.3: A depiction of the setup assumed in the definition of G^1 continuity.

of patches (S-patches) to meet with G^1 continuity [Peters 91].

DEFINITION 4.4.1: S-patches S and T meet with G^1 continuity if and only if there exist functions $\phi(t)$, $\rho(t)$, and $\tau(t)$, $t \in [0, 1]$, such that

I.

$$S(E_p(t)) = T(E_q(t))$$

II.

$$\phi(t)\mathbf{D}_s S(E_p(t)) = \rho(t)\mathbf{D}_i S(E_p(t)) + \tau(t)\mathbf{D}_w T(E_q(t))$$

III.

$$\rho(t)\tau(t) > 0$$

IV.

$$\mathbf{D}_{\dot{s}} S(E_p(t)) \times \mathbf{D}_{\dot{t}} S(E_p(t)) \neq \dot{0}$$

Condition I states that S and T must share a common boundary curve. That is, S and T are continuous (also known as G^0 continuous). Condition II insures that S and T have a continuous tangent plane along their common boundary. This condition is equivalent to $\mathbf{D}_{\dot{s}} S(E_p(t))$, $\mathbf{D}_{\dot{t}} S(E_p(t))$ and $\mathbf{D}_{\dot{w}} T(E_q(t))$ being linearly dependent. Condition III requires S and T to have proper orientation with respect to one another. This condition excludes ‘razor sharp’ joins between patches. Finally, Condition IV requires that the tangent plane shared by S and T is well defined. Without Condition IV, Conditions II and III might be trivially satisfied if one of $\mathbf{D}_{\dot{s}} S(E_p(t))$, $\mathbf{D}_{\dot{t}} S(E_p(t))$ or $\mathbf{D}_{\dot{w}} T(E_q(t))$ were to vanish.

Next, the method used in this thesis to construct a smooth join between a pair of S-patches is shown to satisfy Conditions I and II. Condition III is easily verified after the details for the proposed construction are presented in Chapter 5. Condition IV is not necessarily satisfied by patches constructed using the method proposed in this thesis. Irregularities of the type Condition IV is intended to exclude can occur when control mesh edges degenerate, or when available shape parameters (introduced in Chapter 5) are set to extreme values. It should also be pointed out that regular B-spline surfaces are susceptible to exactly the same problem.

The method of joining S-patches together smoothly involves the construction of an auxiliary surface G , known as an *edge map*, that characterizes the behavior (position and tangent plane) along the proposed join. Let directions \dot{u} and \dot{v} and point o form a coordinate system in the domain of G , and define $G(o + t\dot{u})$, $t \in [0, 1]$ to be the common edge. Since only the behavior of G along the line $o + t\dot{u}$ is of interest, points on this line are represented by their parameter value t . Therefore, $G(o + t\dot{u})$ is denoted simply as $G(t)$. The tangent plane along $G(t)$ is the span of $\mathbf{D}_{\dot{u}} G(t)$ and $\mathbf{D}_{\dot{v}} G(t)$. Chapter 5 will cover the details for constructing the edge maps G ; for now, it can be assumed that such a mapping exists for the common boundary edge.

Given the edge map G , one can construct a pair of patches S and T that meet G^1 simply by interpolating the position and tangent plane of $G(t)$. Interpolating position is trivially accomplished by assigning

$$S(E_p(t)) = T(E_q(t)) = G(t).$$

As a side effect, this assignment implies that derivatives in the direction of the boundary are equal, that is

$$\mathbf{D}_{\dot{s}} S(E_p(t)) = \mathbf{D}_{\dot{r}} T(E_q(t)) = \mathbf{D}_{\dot{u}} G(t). \quad (4.5)$$

Interpolating the tangent plane along G is somewhat more involved. If it were the case that only one pair of patches were being joined along a single edge, then an assignment of the form

$$\mathbf{D}_{\dot{i}} S(E_p(t)) = \mathbf{D}_{-\dot{w}} T(E_q(t)) = \mathbf{D}_{\dot{v}} G(t)$$

would be adequate to interpolate the tangent plane of $G(t)$. However, the intent here is to create G^1 joins along all the edges of a patch. In this more general setting, patches S and T will share a common tangent plane at all points along their common boundary if functions μ_p, ν_p, μ_q and ν_q can be defined such that

$$\mathbf{D}_{\dot{i}} S(E_p(t)) = \mu_p(t) \mathbf{D}_{\dot{u}} G(t) + \nu_p(t) \mathbf{D}_{\dot{v}} G(t), \quad (4.6)$$

and

$$\mathbf{D}_{-\dot{w}} T(E_q(t)) = \mu_q(t) \mathbf{D}_{\dot{u}} G(t) + \nu_q(t) \mathbf{D}_{-\dot{v}} G(t). \quad (4.7)$$

Constructing the functions μ_p and ν_p so that a given patch will meet an adjacent patch smoothly is an important part of this thesis. These functions (determined in Chapter 5) are found subject to constraints that ensure a consistent mixed partial derivative at patch corners.

It is straightforward to show that patches S and T constructed subject to the interpolation conditions outlined in Equations (4.5), (4.6), and (4.7) will meet G^1 . Condition I follows immediately from Equation (4.5). Condition II is satisfied with the following

assignments:

$$\begin{aligned}\phi(t) &= \mu_p(t)\nu_q(t) + \mu_q(t)\nu_p(t), \\ \rho(t) &= \nu_q(t), \\ \tau(t) &= \nu_p(t).\end{aligned}$$

The functions ν_p and ν_q (determined in Chapter 5) are strictly positive for $t \in [0, 1]$, thus satisfying Condition III.

This method of constructing a smooth join between a pair of S-patches is not entirely generally. The specialization results from using edge maps $G(t)$, and functions $\mu(t)$ and $\nu(t)$ that are strictly polynomial (as opposed to rational functions). This means that the differential $\mathbf{D}S(E(t))$ as a function of the edges of the domain polygon of each S-patch S is strictly polynomial. This fact has a rather interesting geometric interpretation: recall that if a depth d S-patch S represents a degree d polynomial, then all of its S-patch panels are affine n -gons (see Section 4.3). If the differential of S is polynomial along the boundary, then the boundary panels are affine n -gons.

Chapter 5

The Generalized B-spline Scheme

The generalized B-spline scheme is fully developed in this chapter. Like the traditional, or regular B-spline surface schemes described in Chapter 3, the new scheme takes as input a control mesh and generates as output a smooth (G^1) composite of surface patches. Unlike regular B-spline surface schemes, the new scheme is not limited to approximating control meshes with a regular rectangular or triangular structure. The bulk of this work consists of a detailed description of the various geometric constructions used to build the approximation to an irregular control mesh. Before presenting these details, more precise definitions of the input and output of the scheme are given.

5.1 Input and Output

The input to the generalized B-spline scheme is a control mesh and the output is a collection of S -patches that meet smoothly along their common boundaries. Conceptually, a control mesh \mathcal{M} may be thought of as a possibly concave polyhedron, with possibly non-planar faces, that may or may not be closed. More formally, a control mesh is defined to be a 3-tuple

$$\mathcal{M} = (\mathbf{V}, \mathbf{E}, \mathbf{F}),$$

where

$\mathbf{V} = \{\mathbf{v} | \mathbf{v} \in M\}$ is a set of *vertices*,

$\mathbf{E} = \{(\mathbf{v}, \mathbf{w}) | \mathbf{v}, \mathbf{w} \in \mathbf{V}\}$ is a set of *edges*,

$\mathbf{F} = \{(\mathbf{v}_1, \dots, \mathbf{v}_n) | (\mathbf{v}_i, \mathbf{v}_{i+1}) \in \mathbf{E}, i = 1, \dots, n \bmod n\}$ is a set of *faces*.

The *order* of a vertex $\mathbf{v} \in \mathbf{V}$, denoted $|\mathbf{v}|$, is defined to be the number of vertices \mathbf{w} , such that (\mathbf{v}, \mathbf{w}) or $(\mathbf{w}, \mathbf{v}) \in \mathbf{E}$. Similarly, the order of a face $f \in \mathbf{F}$, denoted $|f|$, is defined to be the number of vertices (equivalently edges) that f contains.

The control mesh \mathcal{M} is intended to represent a tessellated, oriented 2-manifold (possibly with boundary). Because of this, the sets \mathbf{E} and \mathbf{F} may not be completely arbitrary. For example, each directed edge (\mathbf{v}, \mathbf{w}) can belong to exactly one face. This requirement does not limit the possible surfaces that the scheme can represent; rather it serves to guarantee that a control mesh represents a valid surface. In order to prevent the type of degeneracies Condition IV from the definition of G^1 continuity excludes (see Chapter 4) it is also necessary to require that for all $|\mathbf{v}| \in \mathbf{V}$ and $|f| \in \mathbf{F}$, $|\mathbf{v}| > 2$ and $|f| > 2$.

It is quite possible that the pair $(\mathbf{v}, \mathbf{w}), (\mathbf{w}, \mathbf{v}) \in \mathbf{E}$ belong to distinct faces. Such edges are called *interior edges*. If $(\mathbf{v}, \mathbf{w}) \in \mathbf{E}$ and $(\mathbf{w}, \mathbf{v}) \notin \mathbf{E}$ then (\mathbf{v}, \mathbf{w}) is termed a *boundary edge*. For simplicity, it is assumed throughout the subsequent constructions that all edges are interior edges. This does not mean that a control mesh cannot contain boundary edges; only that constructions involving boundary edges are not defined.

The output of the scheme is a spline surface $\mathcal{S} = (S_i, P_i)$, where S_i is an S -patch defined over a regular polygon P_i , such that

$$\bigcup_i S_i \in G^1.$$

The number of S -patches that form the surface \mathcal{S} is equal to the number of faces in \mathbf{F} when \mathcal{M} is closed. When \mathcal{M} is not closed, fewer S -patches are generated since the constructions are not defined over boundary edges. In this case, one S -patch will be generated for each interior face (an interior face is one consisting only of interior edges).

5.2 Overview of the Scheme

The procedure for converting a control mesh \mathcal{M} (the input) into a smooth spline surface \mathcal{S} (the output) consists of a sequence of affine invariant geometric constructions to determine the control points of the patches $S_i \in \mathcal{S}$, such that $\mathcal{S} \in G^1$ and \mathcal{S} approximates the input control mesh \mathcal{M} . A property of this construction is that \mathcal{S} and \mathcal{M} are of the same topological type. That is, there exists an embedding

$$\Phi : \mathcal{M} \rightarrow \mathcal{S}.$$

While it is not a goal to explicitly build Φ , its existence is assumed so that the intent of the various stages of the constructions is made more apparent. If vertex $\mathbf{v} \in \mathbf{V}$, then the point $\Phi(\mathbf{v})$ is the corresponding point on \mathcal{S} where $|\mathbf{v}|$ patches meet. If $e \in \mathbf{E}$, then $\Phi(e)$ is the boundary curve shared by two surface patches. If $f \in \mathbf{F}$, then $\Phi(f)$ is a surface patch. A consequence of the existence of Φ is that each n -sided S -patch S_i is in 1-to-1 correspondence with a face $f_i \in \mathbf{F}$ where $|f_i| = n$.

The derivation of the generalized B-spline scheme is presented in the next three sections. Conceptually, each of these sections corresponds to finding, respectively, the images of the vertices, edges, and finally the faces of a control mesh under Φ . For each control mesh vertex $\mathbf{v} \in \mathbf{V}$, a piecewise cubic map F is constructed in Bézier triangular form. The purpose of these *vertex maps* is to characterize \mathcal{S} up to second order at the points $\Phi(\mathbf{v})$. Next, a network of boundary curves is constructed. These boundary curves correspond to edges of the control mesh. If $e = (\mathbf{v}_0, \mathbf{v}_1) \in \mathbf{E}$ with vertex maps F_0 and F_1 respectively, then a boundary curve $\Phi(e)$ is found that interpolates second order data of the maps F_0 and F_1 . Similarly, a vector valued function that characterizes a transversal tangent vector along a boundary curve is also found. This function, together with the corresponding boundary curve, is referred to as an *edge map*; it characterizes the position and tangent plane of \mathcal{S} along $\Phi(e)$. Once the edge maps have been constructed for each edge of a face $f \in \mathbf{F}$, an S -patch is found that interpolates these edge maps corresponding to $\Phi(f)$.

The notation in the following sections needs some explanation. Recall that a dot over a symbol is used to distinguish affine vectors from affine points. A superscript is used to order quantities that orbit a vertex; it is always a cyclic index whose value is assumed to be taken modulo the vertex order. Subscripts are used to order quantities over an edge or face, these are typically curve or patch control points. Many quantities will have both superscripts and subscripts; for example, \mathbf{c}_k^i would denote the k^{th} control point of the i^{th} curve about some vertex, \mathbf{b}_j^i would denote the control point \mathbf{b}_j belonging to the i^{th} face about a vertex.

5.3 Vertex Behavior

The first step in the construction is to define, up to second order, the behavior of \mathcal{S} at the points $\Phi(\mathbf{v})$, $\mathbf{v} \in \mathbf{V}$. Conceptually, this behavior is defined by constructing a set of piecewise maps

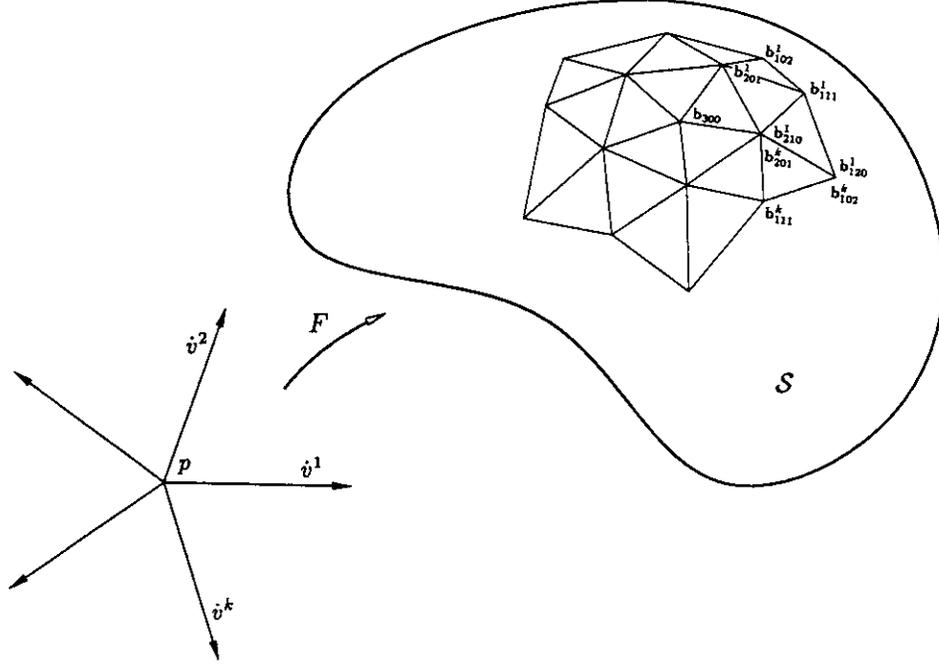
$$F : X \rightarrow M,$$

where X is a 2 dimensional affine space and M is the model space (an affine space of arbitrary dimension). The goal of this section is to construct the maps F in 1-to-1 correspondence with vertices \mathbf{V} . Each map F takes an open neighborhood about a point $p \in X$ onto an open neighborhood about a point $\Phi(\mathbf{v})$ of \mathcal{S} such that $F(p) = \Phi(\mathbf{v})$.

Let $\dot{v}^1, \dots, \dot{v}^k$ be a set of k ($k = |\mathbf{v}|$) distinct directions in X such that each \dot{v}^i lies between \dot{v}^{i-1} and \dot{v}^{i+1} (assume for now that \dot{v}^{i-1} and \dot{v}^{i+1} are independent). Each direction \dot{v}^i is associated with edge $(\mathbf{v}, \mathbf{v}^i) \in \mathbf{E}$, where $\mathbf{v}^1, \dots, \mathbf{v}^k$ are the ordered set of edge sharing neighbors of \mathbf{v} . The directions $\dot{v}^1, \dots, \dot{v}^k$, together with the point p , partition X into k “wedges” (see Figure 5.1). Let $F^i : X \rightarrow M, i = 1, \dots, k$, be a collection of polynomial maps. The composite map F is defined by

$$F(q) = F^i(q) \quad \text{if} \quad |\dot{v}^i \times (q - p)| \geq 0, |(q - p) \times \dot{v}^{i+1}| \geq 0,$$

where \times denotes vector cross product and $q \in X$. The idea is that if a point q lies in the wedge defined by p, \dot{v}^i , and \dot{v}^{i+1} , then $F(q)$ takes on the value of $F^i(q)$. Note that if

Figure 5.1: Vertex map F

$\dot{v}^i \times (q - p) = 0$, then $F(q) = F^i(q) = F^{i+1}(q)$, implying that $F \in C^0$.

Suppose that $F \in C^1$ in an open neighborhood about p . The following proposition demonstrates that not only are the first derivatives of the F^i at p constrained, but certain second derivatives are constrained as well.

PROPOSITION 5.3.1: If $F \in C^1$ in an open neighborhood about p , then

$$\mathbf{D}_{\dot{v}^i, \dot{v}^i}^2 F^i(p) = s^i \mathbf{D}_{\dot{v}^{i-1}, \dot{v}^i}^2 F^{i-1}(p) + t^i \mathbf{D}_{\dot{v}^{i+1}, \dot{v}^i}^2 F^i(p),$$

where

$$\dot{v}^i = s^i \dot{v}^{i-1} + t^i \dot{v}^{i+1}.$$

PROOF : $F \in C^1$ about p implies that there exists a δ such that for all $\epsilon, 0 \leq \epsilon < \delta$, $\mathbf{D}_{\dot{w}} F^i(p + \epsilon \dot{v}^i) = \mathbf{D}_{\dot{w}} F^{i-1}(p + \epsilon \dot{v}^i)$, for all $\dot{w} \in X$. Therefore taking $\dot{w} = \dot{v}^i$

$$\mathbf{D}_{\dot{v}^i} F^i(p + \epsilon \dot{v}^i) = s^i \mathbf{D}_{\dot{v}^{i-1}} F^{i-1}(p + \epsilon \dot{v}^i) + t^i \mathbf{D}_{\dot{v}^{i+1}} F^i(p + \epsilon \dot{v}^i).$$

Differentiating this equation with respect to ϵ and evaluating at $\epsilon = 0$ gives the desired result. \square

Proposition 5.3.1 indicates an important relationship that must exist among polynomial patches (or any C^2 functions) that meet C^1 . A collection of patches that meet C^1 at a point, and satisfy Proposition 5.3.1, are said to be *twist compatible*.

To summarize, in order to construct the composite map F to be twist compatible, $F^i, i = 1, \dots, k$ must satisfy:

$$F^{i-1}(p) = F^i(p), \quad (5.1)$$

$$\mathbf{D}_{\dot{v}^i} F^i(p) = s^i \mathbf{D}_{\dot{v}^{i-1}} F^{i-1}(p) + t^i \mathbf{D}_{\dot{v}^{i+1}} F^i(p), \quad (5.2)$$

$$\mathbf{D}_{\dot{v}^i, \dot{v}^i}^2 F^i(p) = s^i \mathbf{D}_{\dot{v}^{i-1}, \dot{v}^i}^2 F^{i-1}(p) + t^i \mathbf{D}_{\dot{v}^{i+1}, \dot{v}^i}^2 F^i(p), \quad (5.3)$$

where

$$\dot{v}^i = s^i \dot{v}^{i-1} + t^i \dot{v}^{i+1}.$$

In many surface interpolation problems where a boundary curve network is given a priori, Constraints (5.1) and (5.2) can be satisfied by any collection of patches that interpolate the curve network (assuming the curves of the network share a common position and tangent plane at the vertices). Satisfying Constraint (5.3) involves solving a corresponding $k \times k$ ($k = |\mathbf{v}|$) linear system of equations. Unfortunately, this system is in general non-singular only when k is odd. Therefore, constructing boundary curves a priori will not work in general, since polynomial patches cannot always be found that satisfy Constraint (5.3).

One way to avoid this difficulty, is to construct the boundary curve network to interpolate C^2 data at the vertices [Peters 91]. This is sufficient, but not necessary. A more general approach is taken in this thesis that involves first constructing the *twist vectors* $\mathbf{D}_{\dot{v}^i, \dot{v}^{i+1}}^2 F^i(p)$, then using Constraint (5.3) to determine $\mathbf{D}_{\dot{v}^i, \dot{v}^i}^2 F^i(p)$. Once these derivatives have been determined, boundary curves and transversal vector fields are constructed that interpolate the second order data of the vertex maps.

5.3.1 Vertex Maps in Bézier Triangular Form

Instead of constructing the vertex maps F^i , $i = 1, \dots, k$ in Taylor form (i.e. finding $F^i(p)$, $\mathbf{D}_{\dot{v}^i} F^i(p)$, $\mathbf{D}_{\dot{v}^i, \dot{v}^{i+1}}^2 F^i(p)$, and $\mathbf{D}_{\dot{v}^i, \dot{v}^i}^2 F^i(p)$, $i = 1, \dots, k$), these maps are found in the more geometrically intuitive Bernstein form by constructing Bézier control points. The geometric relationships that exist between Bézier control points are generally much easier to visualize and comprehend than are the relationships between the Taylor coefficients which are derivatives of various orders at p .

Each F^i will be represented as a degree 3 Bézier triangle (see Section 2.5) with domain triangle $p, p + \dot{v}^i, p + \dot{v}^{i+1}$, and control points $\mathbf{b}_{\vec{i}}^i$, where $|\vec{i}| = 3$, and $i = 1, \dots, k$ (see Figure 5.1). Since $F \in C^0$, it must be that

$$\mathbf{b}_{300}^{i-1} = \mathbf{b}_{300}^i, \quad (5.4)$$

$$\mathbf{b}_{201}^{i-1} = \mathbf{b}_{210}^i, \quad (5.5)$$

$$\mathbf{b}_{102}^{i-1} = \mathbf{b}_{120}^i, \quad (5.6)$$

$$\mathbf{b}_{003}^{i-1} = \mathbf{b}_{030}^i. \quad (5.7)$$

The following identities, based on formulae for a Bézier triangular patch, provide the relationships between the Taylor and Bézier forms of the maps F^i :

$$F^i(p) = \mathbf{b}_{300}^i, \quad (5.8)$$

$$\mathbf{D}_{\dot{v}^i} F^i(p) = 3(\mathbf{b}_{210}^i - \mathbf{b}_{300}^i), \quad (5.9)$$

$$\mathbf{D}_{\dot{v}^i, \dot{v}^i}^2 F^i(p) = 6(\mathbf{b}_{120}^i - 2\mathbf{b}_{210}^i + \mathbf{b}_{300}^i), \quad (5.10)$$

$$\mathbf{D}_{\dot{v}^i, \dot{v}^{i+1}}^2 F^i(p) = 6(\mathbf{b}_{111}^i - \mathbf{b}_{210}^i - \mathbf{b}_{201}^i + \mathbf{b}_{300}^i). \quad (5.11)$$

Using these relationships, Constraints (5.1) through (5.3) may be rewritten

$$\mathbf{b}_{300}^i = \mathbf{b}_{300}^{i+1}, \quad (5.12)$$

$$\mathbf{b}_{210}^i = r^i \mathbf{b}_{300}^{i+1} + s^i \mathbf{b}_{210}^{i-1} + t^i \mathbf{b}_{210}^{i+1}, \quad (5.13)$$

$$\mathbf{b}_{120}^i = r^i \mathbf{b}_{210}^{i+1} + s^i \mathbf{b}_{111}^{i-1} + t^i \mathbf{b}_{111}^{i+1}, \quad (5.14)$$

where $r^i = 1 - s^i - t^i$. Note that the points $\mathbf{b}_{030}^i, \mathbf{b}_{021}^i, \mathbf{b}_{012}^i$, and $\mathbf{b}_{003}^i, i = 1, \dots, k$, do not affect the second order behavior of $F(p)$. Therefore, these points need not be specified. This seems to indicate that constructing the F^i as quadratic, rather than cubic, triangular patches would be sufficient. The reason for choosing cubics has to do with generalizing the constructions of B-spline surface. These constructions have a much “cleaner” geometric interpretation if the vertex maps are assumed to be cubic. This should become evident in the next section.

5.3.2 Constructing \mathbf{b}_{111}

The first step in computing the Bézier representation of the maps F^i is to construct the control points \mathbf{b}_{111}^i . In [Loop & DeRose 90], these points were referred to as *key points*. This name reflected the importance of these points in influencing the shape of the resulting surface, as well as their pivotal role in constructing the remaining Bézier control points of the maps F^i . A more fitting name in keeping with the CAGD literature might have been *twist points*, due their close relationship to the *twist vectors* or mixed partial derivatives.

A great deal of latitude exists in determining the positions of the \mathbf{b}_{111}^i . These points could reasonably be left as undetermined shape parameters subject to user specification. However, this approach would place an unnecessary burden on users, requiring them to make a multitude of decisions to design even simple shapes. Instead, reasonable default positions are given. These points can subsequently be manipulated if the resulting shape is not acceptable.

As a default, the points \mathbf{b}_{111}^i are constructed to lie on control mesh faces, $f \in \mathbf{F}$. If $|f| = n$, then n key points are constructed to lie on the surface of face f . This surface is defined by considering the vertices of f as control points of a depth 1 S-patch. Let

$$q_j = \frac{1}{3}p_{j-1} + \frac{1}{3}p_j + \frac{1}{3}p_{j+1},$$

be a point inside an n -sided domain polygon p_1, \dots, p_n . Let $f = (\mathbf{v}_1, \dots, \mathbf{v}_n)$, and let ℓ_1, \dots, ℓ_n be the component functions of the S-patch embedding map (see Section 4.1).

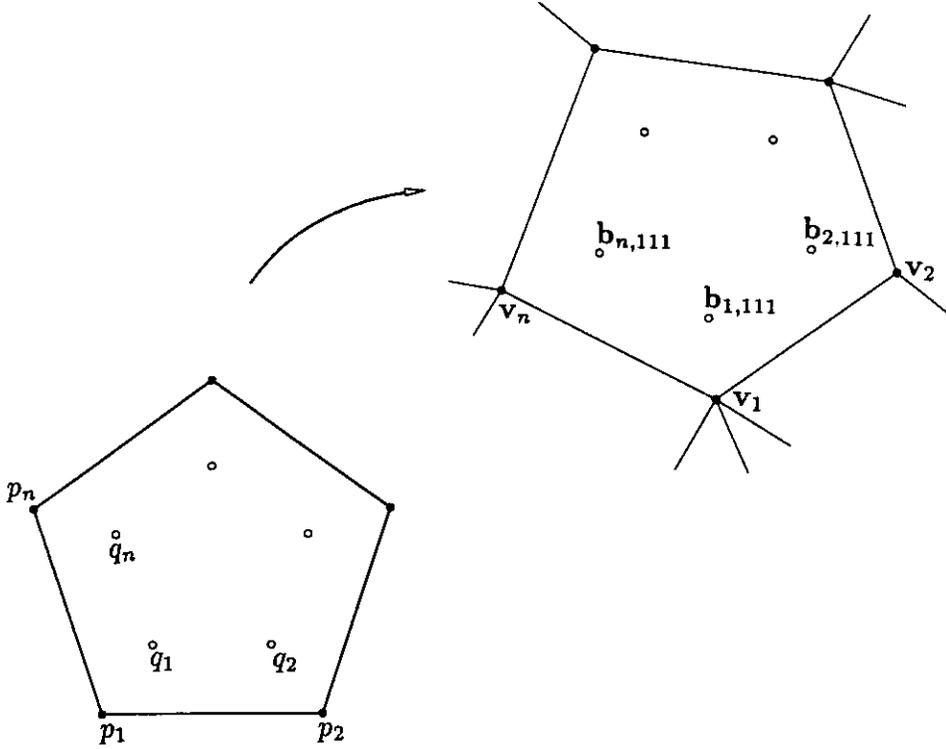


Figure 5.2: The construction of keypoints $\mathbf{b}_{1,111}, \dots, \mathbf{b}_{n,111}$ on a single face

The key points associated with face f are constructed by

$$\mathbf{b}_{j,111} = \ell_1(q_j)\mathbf{v}_1 + \dots + \ell_n(q_j)\mathbf{v}_n, \quad j = 1, \dots, n. \quad (5.15)$$

This construction is illustrated in Figure 5.2. Each keypoint is uniquely associated with a vertex-face pair. Note that relative to its defining face, the key point is indexed by a subscript, relative to the vertex it orbits, it would be indexed by a superscript.

The positioning of key points is an important parameter that influences the shape of the surface. Many possibilities exist and it should be understood that the proposed values have no special significance other than being simple to describe. Empirical tests have shown reasonable results. More insight into defining values for key points is needed.

5.3.3 Constructing \mathbf{b}_{300}

The next step is to construct the points \mathbf{b}_{300}^i . These points lie on the surface \mathcal{S} and coincide with the points $\Phi(\mathbf{v})$. Constraint (5.4) indicates that $\mathbf{b}_{300}^{i-1} = \mathbf{b}_{300}^i, i = 1, \dots, k$; thus the superscript i is dropped from \mathbf{b}_{300}^i .

The point \mathbf{b}_{300} is found as the point that breaks the line containing the control mesh vertex \mathbf{v} and the centroid of the key points that orbit \mathbf{v} into a ratio of $1 - \alpha : \alpha$, where α is scalar weighting factor to be determined shortly. This construction is expressed as

$$\mathbf{b}_{300} = \alpha \mathbf{v} + \frac{1-\alpha}{k} \mathbf{b}_{111}^1 + \dots + \frac{1-\alpha}{k} \mathbf{b}_{111}^k, \quad (5.16)$$

where $\mathbf{b}_{111}^1, \dots, \mathbf{b}_{111}^k$ are the keypoints orbiting \mathbf{v} constructed in the previous section. Construction (5.16) is illustrated in Figure 5.3.

The weighting factor α can be used as a shape parameter, i.e. as $\alpha \rightarrow 1$, the surface gets closer to \mathbf{v} ; as $\alpha \rightarrow 0$, the surface gets closer to the centroid of the keypoints. While this extra freedom is potentially useful, it will be ignored in the present scheme. Instead, a value of α is determined that generalizes the constructions of bicubic tensor product, and quartic triangular B-splines. This is achieved by setting

$$\alpha = \frac{1}{2} \cos \frac{2\pi}{k},$$

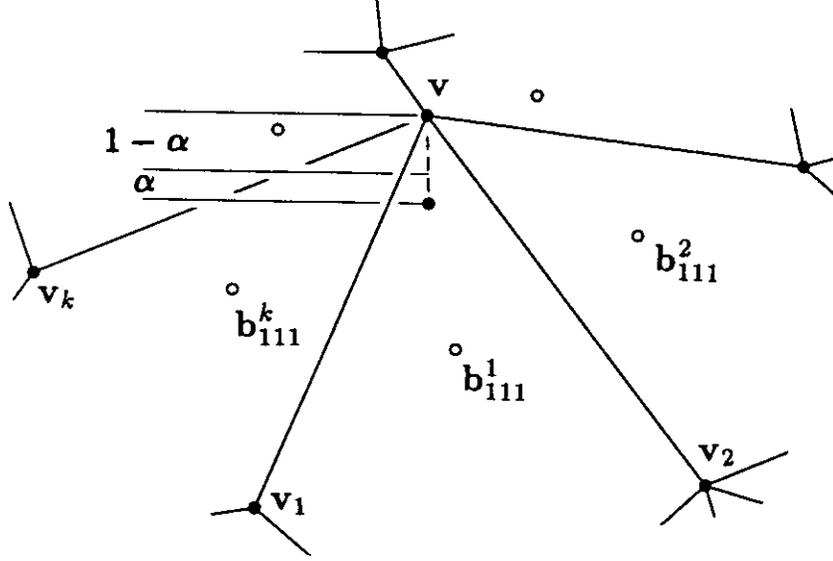
where $k = |\mathbf{v}|$. Empirically, this value of α gives good results and has been used in all examples throughout this thesis.

5.3.4 Constructing \mathbf{b}_{210}

Next, the points \mathbf{b}_{210}^i are constructed subject to Constraint (5.13). This constraint requires that the points $\mathbf{b}_{210}^i, i = 1, \dots, k$, together with the point \mathbf{b}_{300} be coplanar¹. In fact, the plane spanned by these points will be the tangent plane of \mathcal{S} at $\Phi(\mathbf{v})$.

Although Constraints (5.12) through (5.14) allow a very general relationship to exist among the Bézier control points of a vertex map, this generality is not fully exploited

¹Coplanarity is necessary but not sufficient to satisfy Constraint (5.13)

Figure 5.3: The construction of the point \mathbf{b}_{300}

here. Instead, it is assumed that the points $\mathbf{b}_{210}^1, \dots, \mathbf{b}_{210}^k$ form an affine k -gon with centroid \mathbf{b}_{300} . This fact will lead to many subsequent simplifications, and is the basis of the following construction:

$$\mathbf{b}_{210}^i = \alpha \mathbf{v} + m_{i-1} \mathbf{b}_{111}^1 + m_{i-2} \mathbf{b}_{111}^2 + \dots + m_{i-k} \mathbf{b}_{111}^k, \quad (5.17)$$

where

$$m_j = \frac{1}{k} [1 - \alpha + \beta (\cos \frac{2\pi j}{k} - \tan \frac{\pi}{k} \sin \frac{2\pi j}{k})], \quad (5.18)$$

and where β is another scalar weighting factor. The weighting factor β can be used as a shape parameter that controls the magnitude of tangent vectors about the point $\Phi(\mathbf{v})$. However, for the remainder of this work it will be assumed that $beta = 1$. The parameter β is included here merely to highlight its existence.

The following claim establishes formally that Construction (5.17) satisfies Constraint (5.13).

CLAIM 5.3.1: If $\mathbf{b}_{210}^i, i = 1, \dots, k$, are found by Construction (5.17), then

$$\mathbf{b}_{210}^i = r \mathbf{b}_{300} + s \mathbf{b}_{210}^{i-1} + t \mathbf{b}_{210}^{i+1},$$

where

$$r = 1 - \sec \frac{2\pi}{k}, \quad s = \frac{1}{2} \sec \frac{2\pi}{k}, \quad t = \frac{1}{2} \sec \frac{2\pi}{k}.$$

PROOF : Taking advantage of standard trigonometric identities, it follows that

$$\begin{aligned} r\mathbf{b}_{300}^i + s\mathbf{b}_{210}^{i-1} + t\mathbf{b}_{210}^{i+1} &= \alpha\mathbf{v} + \sum_{j=1}^k [r \frac{1-\alpha}{k} + s m_{i-j-1} + t m_{i-j+1}] \mathbf{b}_{111}^j, \\ &= \alpha\mathbf{v} + \sum_{j=1}^k \frac{1}{k} [1 - \alpha + \beta \sec \frac{2\pi}{k} ((\cos \frac{2\pi(i-j-1)}{k} + \cos \frac{2\pi(i-j+1)}{k}) \\ &\quad - \tan \frac{\pi}{k} (\sin \frac{2\pi(i-j-1)}{k} + \frac{\pi}{k} \sin \frac{2\pi(i-j+1)}{k}))] \mathbf{b}_{111}^j, \\ &= \alpha\mathbf{v} + \sum_{j=1}^k \frac{1}{k} [1 - \alpha + \beta (\cos \frac{2\pi(i-j)}{k} - \tan \frac{\pi}{k} \sin \frac{2\pi(i-j)}{k})] \mathbf{b}_{111}^j, \\ &= \alpha\mathbf{v} + \sum_{j=1}^k m_{i-j} \mathbf{b}_{111}^j, \\ &= \mathbf{b}_{210}^i. \end{aligned}$$

□

Thus, if $\dot{v}^i = \frac{1}{2} \sec \frac{2\pi}{k} (\dot{v}^{i-1} + \dot{v}^{i+1})$, $i = 1, \dots, k$, then Constraint (5.17) is satisfied. It is important to point out that when $k = 4$ (hence $\cos \frac{\pi}{2} = 0$), this expression is *not* well defined. This exception manifests itself in the next section as a degree of freedom in determining the boundary curves.

5.3.5 Constructing \mathbf{b}_{120}

Finally, the points \mathbf{b}_{120}^i are constructed. These points must be constructed subject to Constraint (5.14). Since the points \mathbf{b}_{111}^i and \mathbf{b}_{210}^i , and scalars r, s , and t have been determined, it follows immediately from Constraint (5.14) that

$$\mathbf{b}_{120}^i = (1 - \sec \frac{2\pi}{k}) \mathbf{b}_{210}^i + \frac{1}{2} \sec \frac{2\pi}{k} \mathbf{b}_{111}^{i-1} + \frac{1}{2} \sec \frac{2\pi}{k} \mathbf{b}_{111}^i. \quad (5.19)$$

The purpose of constructing the vertex maps F^i , $i = 1, \dots, k$, is to partially define the spline surface \mathcal{S} (in open neighborhoods about the points $\Phi(\mathbf{v})$). The data defined by

these maps is next interpolated to form a network of boundary curves and transversal vector fields.

5.4 Edge Behavior

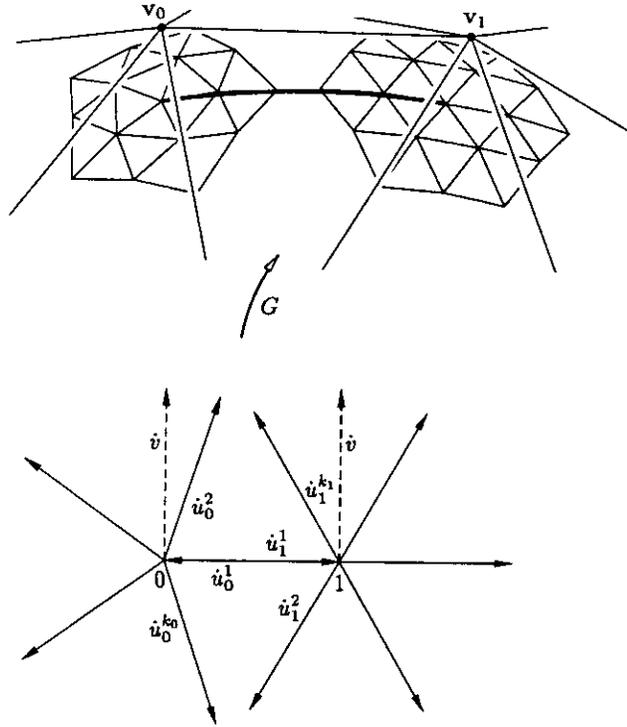
The data contained in the vertex maps defined in the previous section are now interpolated to form a network of boundary curves and transversal vector fields. These functions are combined in a single map

$$G : X \rightarrow M,$$

where X is a 2 dimensional affine space and M is an affine space of arbitrary dimension. This *edge map* characterizes the first order behavior of $\Phi(e), e \in \mathbf{E}$. Strictly speaking, the parameter of the function G is a point from the 2 dimensional affine space X ; however, only values of G and its differential $\mathbf{D}G$ restricted to a 1 dimensional subset of X are important here. Therefore, in the interest of preventing excessively cumbersome notation, the bivariate function G will be abbreviated so that the parameter to G is a real number. Thus $G(u)$ should be interpreted as $G(u) = G((1 - u)u_0 + u u_1), u \in [0, 1]$ and $u_0, u_1 \in X$. With this notation $G(u)$ is a curve, and $\mathbf{D}_i G(u)$ is a transversal vector field.

Let $(\mathbf{v}_0, \mathbf{v}_1)$ be an edge belonging to some face $f \in \mathbf{G}$, and let F_0 and F_1 be the vertex maps corresponding to \mathbf{v}_0 and \mathbf{v}_1 respectively. Assume that v_0^1 is the direction in the domain of F_0 that corresponds to edge $(\mathbf{v}_0, \mathbf{v}_1)$; similarly, assume v_1^1 is the direction in the domain of F_1 that corresponds to edge $(\mathbf{v}_1, \mathbf{v}_0)$. Define directions $\dot{u}_0^1, \dots, \dot{u}_0^{k_0}$ and $\dot{u}_1^1, \dots, \dot{u}_1^{k_0}$ in the domain of G such that $\dot{u}_0^i = \frac{1}{2} \sec \frac{2\pi}{k_0} (\dot{u}_0^{i-1} + \dot{u}_0^{i+1})$ and $\dot{u}_1^i = \frac{1}{2} \sec \frac{2\pi}{k_1} (\dot{u}_1^{i-1} + \dot{u}_1^{i+1})$, where $k_0 = |\mathbf{v}_0|$ and $k_1 = |\mathbf{v}_1|$. Geometrically, the end points of the vectors $\dot{u}_0^1, \dots, \dot{u}_0^{k_0}$ and $\dot{u}_1^1, \dots, \dot{u}_1^{k_0}$ form the affine images of a regular k_0 -gon and k_1 -gon respectively. Note that $\dot{u}_0^1 = -\dot{u}_1^1$. These definitions are illustrated in Figure 5.4.

The idea is to construct the edge map G so that at endpoint u_0 , G interpolates the vertex map F_0 , and at u_1 , G interpolates F_1 . This means that in a neighborhood about

Figure 5.4: Edge map G

u_0 , G is interpreted as

$$G = F_0 \circ U_0,$$

where U_0 is the unique affine map such that

$$U_0 : v_0 \rightarrow u_0, \quad U_0 : v_0^i \rightarrow u_0^i, \quad i = 1, \dots, k_0.$$

A similar interpretation exists about the other endpoint u_1 .

This idea is made explicit by making the following assignment among the discrete

quantities

$$\begin{aligned}
G(0) &= F_0(v_0), & G(1) &= F_1(v_1), \\
\mathbf{D}_{\dot{u}_0^1} G(0) &= \mathbf{D}_{\dot{v}_0^1} F_0^1(v_0), & \mathbf{D}_{\dot{u}_1^1} G(1) &= \mathbf{D}_{\dot{v}_1^1} F_1^1(v_1), \\
\mathbf{D}_{\dot{u}_0^2} G(0) &= \mathbf{D}_{\dot{v}_0^2} F_0^1(v_0), & \mathbf{D}_{\dot{u}_1^{k_1}} G(1) &= \mathbf{D}_{\dot{v}_1^{k_1}} F_1^1(v_1), \\
\mathbf{D}_{\dot{u}_0^1, \dot{u}_0^1}^2 G(0) &= \mathbf{D}_{\dot{v}_0^1, \dot{v}_0^1}^2 F_0^1(v_0), & \mathbf{D}_{\dot{u}_1^1, \dot{u}_1^1}^2 G(1) &= \mathbf{D}_{\dot{v}_1^1, \dot{v}_1^1}^2 F_1^1(v_1), \\
\mathbf{D}_{\dot{u}_0^1, \dot{u}_0^2}^2 G(0) &= \mathbf{D}_{\dot{v}_0^1, \dot{v}_0^2}^2 F_0^1(v_0), & \mathbf{D}_{\dot{u}_1^1, \dot{u}_1^{k_1}}^2 G(1) &= \mathbf{D}_{\dot{v}_1^1, \dot{v}_1^{k_1}}^2 F_1^1(v_1).
\end{aligned}$$

It remains to determine the values of $G(u)$ and $\mathbf{D}G(u)$, $0 < u < 1$. This procedure is quite straightforward: $G(u)$ and $\mathbf{D}_v G(u)$ are found as the minimum degree polynomials that interpolate the data at the endpoints. The details appear in the next two sections.

5.4.1 Boundary Curves

In the general case, $G(u)$ is found to be the unique quintic polynomial that interpolates the vertex data at the endpoints of G . Let

$$G(u) = \sum_{i=0}^5 \mathbf{c}_i B_i^5(u),$$

where $\mathbf{c}_0, \dots, \mathbf{c}_5$ are Bézier control points. Let $\mathbf{b}_{0,\vec{i}}$ and $\mathbf{b}_{1,\vec{i}}$ be the Bézier control points of the vertex maps F_0^1 and $F_1^{k_1}$ respectively. The Bézier control points of $G(u)$ can be determined from the vertex map control points as follows

$$\begin{aligned}
\mathbf{c}_0 &= \mathbf{b}_{0,300}, \\
\mathbf{c}_1 &= \frac{2}{5}\mathbf{b}_{0,300} + \frac{3}{5}\mathbf{b}_{0,210}, \\
\mathbf{c}_2 &= \frac{1}{10}\mathbf{b}_{0,300} + \frac{6}{10}\mathbf{b}_{0,210} + \frac{3}{10}\mathbf{b}_{0,120}, \\
\mathbf{c}_3 &= \frac{1}{10}\mathbf{b}_{1,300} + \frac{6}{10}\mathbf{b}_{1,201} + \frac{3}{10}\mathbf{b}_{1,102}, \\
\mathbf{c}_4 &= \frac{2}{5}\mathbf{b}_{1,300} + \frac{3}{5}\mathbf{b}_{1,201}, \\
\mathbf{c}_5 &= \mathbf{b}_{1,300}.
\end{aligned}$$

Recall that control points \mathbf{b}_{120} are not constrained when $k = 4$. Therefore, the number of constraints on a boundary curve is reduced by one for each order 4 vertex.

This property can be used to reduce the polynomial degree of a boundary curve. If the vertices at both endpoints are order 4, then the boundary curve need only be cubic. If only one vertex is order 4, then the boundary curve need only be quartic. Otherwise in the general case, the boundary curve must be quintic. This freedom can be dealt with by computing the free control point \mathbf{b}_{120} from other known control points. If $|\mathbf{v}_0| = |\mathbf{v}_1| = 4$ then

$$\mathbf{b}_{0,120} = \mathbf{b}_{1,201}.$$

If $|\mathbf{v}_0| = 4$ and $|\mathbf{v}_1| \neq 4$, then set

$$\mathbf{b}_{0,120} = \mathbf{b}_{1,201} + \mathbf{b}_{1,102} - \mathbf{b}_{0,201}.$$

These formulae are derived uniquely since all degrees of freedom are removed by requiring lower degree boundary curves. Once $\mathbf{b}_{0,120}$ is computed, the above formulae for $\mathbf{c}_0, \dots, \mathbf{c}_5$ will suffice for all boundary curves.

5.4.2 Boundary Differential

Once the boundary curves have been determined, the next step is to construct the differential $\mathbf{D}G(u)$ along each boundary. For each fixed value of u , the differential of $G(u)$ is a linear map

$$\mathbf{D}G(u) : X \rightarrow M$$

on vectors of X . Let the vectors \dot{u} and \dot{v} be orthonormal (i.e. $\dot{u} \cdot \dot{v} = 0$, $\dot{u} \cdot \dot{u} = 1$ and $\dot{v} \cdot \dot{v} = 1$). Since $G(u)$ is known, it's derivative $\mathbf{D}_{\dot{u}}G(u)$ is also known. Therefore, $\mathbf{D}G(u)$ will be completely determined once $\mathbf{D}_{\dot{v}}G(u)$ is known. The function $\mathbf{D}_{\dot{v}}G(u)$ is equivalent to the transversal vector field and is found so that the data defined by F_0 and F_1 at the endpoints is interpolated.

Since the points $u_0 + \dot{u}_0^i$, $i = 1, \dots, k_0$ form a regular k_0 -gon, the relationship between \dot{u}, \dot{v} and \dot{u}_0^{i+1} can be expressed

$$\dot{u}_0^{i+1} = \cos \frac{2\pi i}{k_0} \dot{u} + \sin \frac{2\pi i}{k_0} \dot{v}.$$

By linearity of differentiation it follows that

$$\mathbf{D}_{\dot{u}_0^2} G(0) = \cos \frac{2\pi}{k_0} \mathbf{D}_{\dot{u}} G(0) + \sin \frac{2\pi}{k_0} \mathbf{D}_{\dot{v}} G(0).$$

Therefore the differential of G in the direction of \dot{v} , evaluated at zero can be expressed

$$\mathbf{D}_{\dot{v}} G(0) = -\cot \frac{2\pi}{k_0} \mathbf{D}_{\dot{u}_0^1} G(0) + \csc \frac{2\pi}{k_0} \mathbf{D}_{\dot{u}_0^2} G(0). \quad (5.20)$$

By a similar argument,

$$\mathbf{D}_{\dot{v}} G(1) = -\cot \frac{2\pi}{k_1} \mathbf{D}_{\dot{u}_1^1} G(1) + \csc \frac{2\pi}{k_1} \mathbf{D}_{\dot{u}_1^{k_1}} G(1). \quad (5.21)$$

Equations (5.20) and (5.21) are differentiated with respect to \dot{u} to obtain

$$\mathbf{D}_{\dot{u}, \dot{v}}^2 G(0) = -\cot \frac{2\pi}{k_0} \mathbf{D}_{\dot{u}_0^1, \dot{u}_0^1}^2 G(0) + \csc \frac{2\pi}{k_0} \mathbf{D}_{\dot{u}_0^1, \dot{u}_0^2}^2 G(0), \quad (5.22)$$

$$\mathbf{D}_{\dot{u}, \dot{v}}^2 G(1) = -\cot \frac{2\pi}{k_1} \mathbf{D}_{\dot{u}_1^1, \dot{u}_1^1}^2 G(1) + \csc \frac{2\pi}{k_1} \mathbf{D}_{\dot{u}_1^1, \dot{u}_1^{k_1}}^2 G(1). \quad (5.23)$$

Since $\mathbf{D}_{\dot{v}} G(0)$, $\mathbf{D}_{\dot{u}, \dot{v}}^2 G(0)$, $\mathbf{D}_{\dot{u}, \dot{v}}^2 G(1)$, and $\mathbf{D}_{\dot{v}} G(1)$ are known, $\mathbf{D}_{\dot{v}} G(u)$ can be found as a cubic. Let

$$\mathbf{D}_{\dot{v}} G(u) = \sum_{i=0}^3 \dot{\mathbf{c}}_i B_i^3(u),$$

be the representation of $\mathbf{D}_{\dot{v}} G(u)$ in Bernstein form, where $\dot{\mathbf{c}}$ are Bézier *control vectors*. Combining Equations (5.20) through (5.23) and Equations (5.9) through (5.11) along with the definition of G 's endpoint behavior results in the following formulae:

$$\dot{\mathbf{c}}_0 = 3r_0 \mathbf{b}_{0,300} + 3s_0 \mathbf{b}_{0,210} + 3t_0 \mathbf{b}_{0,201}, \quad (5.24)$$

$$\dot{\mathbf{c}}_1 = r_0 (\mathbf{b}_{0,300} + 2\mathbf{b}_{0,210}) + s_0 (\mathbf{b}_{0,210} + 2\mathbf{b}_{0,120}) + t_0 (\mathbf{b}_{0,201} + 2\mathbf{b}_{0,111}), \quad (5.25)$$

$$\dot{\mathbf{c}}_2 = r_1 (\mathbf{b}_{1,300} + 2\mathbf{b}_{1,201}) + s_1 (\mathbf{b}_{1,201} + 2\mathbf{b}_{1,102}) + t_1 (\mathbf{b}_{1,210} + 2\mathbf{b}_{1,111}), \quad (5.26)$$

$$\dot{\mathbf{c}}_3 = 3r_1 \mathbf{b}_{1,300} + 3s_1 \mathbf{b}_{1,201} + 3t_1 \mathbf{b}_{1,210}, \quad (5.27)$$

where

$$r_j = \cot \frac{2\pi}{k_j} - \csc \frac{2\pi}{k_j}, \quad s_j = -\cot \frac{2\pi}{k_j}, \quad t_j = \csc \frac{2\pi}{k_j}, \quad j = 0, 1.$$

Determining $G(u)$ and $\mathbf{D}_{\dot{v}} G(u)$ for each pair of adjacent vertex maps completely specifies a boundary curve network with associated transversal vector field. These functions interpolate the data from the vertex maps. In the next section, these edge maps are interpolated to form patches.

5.5 Face Behavior

The behavior of the surface \mathcal{S} corresponding to an entire face $f \in \mathbf{F}$ can now be determined. This is accomplished by finding a map

$$H : X \rightarrow M$$

where X is a 2 dimensional affine space and M is an affine space of arbitrary dimension. The idea is that the mapping H will take a domain polygon $P \subset X$ onto $\Phi(f)$, $f \in \mathbf{F}$. The mapping H is found in S-patch form by interpolating a set of edge maps computed in the previous section.

Let the face $f = (\mathbf{v}_1, \dots, \mathbf{v}_n)$, and let G_i be the edge map for $(\mathbf{v}_i, \mathbf{v}_{i+1}) \in \mathbf{E}$, $i = 1, \dots, n$, as computed in Section 5.4. Let $p_1, \dots, p_n \in X$ be the vertices of the regular n -gon P , let \dot{t}_i be the vector $p_{i+1} - p_i$, and let E_i denote the i^{th} edge of P , that is, $E_i(t) = (1-t)p_i + t p_{i+1}$, $t \in [0, 1]$. The S-patch H will interpolate the boundary curves $G_1(t), \dots, G_n(t)$ by requiring that

$$H(E_i(t)) = G_i(t), \quad t \in [0, 1], i = 1, \dots, n. \quad (5.28)$$

Interpolating the tangent planes spanned by $\mathbf{D} G_i(t)$ is considerably more difficult.

Simply equating the differentials $\mathbf{D} H(E_i(t))$ and $\mathbf{D} G_i(t)$ will not suffice, since in general $\mathbf{D} G_{i-1}(1) \neq \mathbf{D} G_i(0)$. The solution is to expand $\mathbf{D} H(E_i(t))$ as a pair of functions $\mathbf{D}_{\dot{t}_i} H(E_i(t))$ and $\mathbf{D}_{-\dot{t}_{i-1}} H(E_i(t))$ that together characterize the differential of H along the perimeter of P .

It follows from Equation (5.28) that in boundary directions

$$\mathbf{D}_{\dot{t}_i} H(E_i(t)) = \mathbf{D}_{\dot{u}_i} G_i(t), \quad t \in [0, 1]. \quad (5.29)$$

In this equation, \dot{t}_i and \dot{u}_i are directions in the domains of H and G_i respectively along a boundary corresponding to the parameter t . Defining $\mathbf{D} H(E_i(t))$ in directions transversal to a boundary is done by finding functions μ_i, ν_i over $[0, 1]$ such that

$$\mathbf{D}_{-\dot{t}_{i-1}} H(E_i(t)) = \mu_i(t) \mathbf{D}_{\dot{u}_i} G_i(t) + \nu_i(t) \mathbf{D}_{\dot{v}_i} G_i(t). \quad (5.30)$$

The functions μ_i and ν_i are found as polynomials subject to constraints. Once these functions have been determined, $H(E_i(t))$ and $\mathbf{D}H(E_i(t))$ are used to determine the S-patch boundary panels of H . Finally, the remaining behavior of H on the interior of P is determined.

5.5.1 Determining the Functions μ and ν

The functions μ_i and ν_i are found by setting up a system of constraints involving the values and derivatives of the edge maps $G_i, i = 1, \dots, n$, at the endpoints 0 and 1. Since the edge maps G_i interpolate common vertex map data at endpoints, it follows that

$$G_{i-1}(1) = G_i(0), \quad (5.31)$$

$$\mathbf{D}_{-\dot{u}_{i-1}} G_{i-1}(1) = \mathbf{D}_{\dot{s}_i} G_i(0), \quad (5.32)$$

$$\mathbf{D}_{\dot{r}_{i-1}} G_{i-1}(1) = \mathbf{D}_{\dot{u}_i} G_i(0), \quad (5.33)$$

$$\mathbf{D}_{-\dot{u}_{i-1}, \dot{r}_{i-1}}^2 G_{i-1}(1) = \mathbf{D}_{\dot{u}_i, \dot{s}_i}^2 G_i(0), \quad (5.34)$$

where

$$\dot{s}_i = \alpha_i \dot{u}_i + \beta_i \dot{v}_i, \quad (5.35)$$

$$\dot{r}_i = -\alpha_{i+1} \dot{u}_i + \beta_{i+1} \dot{v}_i. \quad (5.36)$$

The scalars α_i and β_i are known from Section 5.4 to be

$$\alpha_i = \cos \frac{2\pi}{k_i}, \quad \beta_i = \sin \frac{2\pi}{k_i},$$

where $k_i = |\mathbf{v}_i|$, $i = 1, \dots, n$. In order maintain generality, all that is assumed about α_i and β_i is that they exist so that Conditions (5.31) through (5.34) are satisfied.

The first set of constraints are found by setting $t_i = 0$ in Equation (5.30), and using Equation (5.29) and Condition (5.31) to show that

$$\mathbf{D}_{\dot{s}_i} G_i(0) = \mu_i(0) \mathbf{D}_{\dot{u}_i} G_i(0) + \nu_i(0) \mathbf{D}_{\dot{v}_i} G_i(0).$$

Therefore, from Equation (5.35) it follows that

$$\mu_i(0) = \alpha_i, \quad \nu_i(0) = \beta_i, \quad i = 1, \dots, n.$$

The second set of constraints are found by observing that since P is a regular n -gon, it can be shown that

$$-\dot{t}_{i-1} = -2 \cos \frac{2\pi}{n} \dot{t}_i + \dot{t}_{i+1}.$$

Therefore, by linearity of differentiation

$$\mathbf{D}_{-\dot{t}_{i-1}} H = -2 \cos \frac{2\pi}{n} \mathbf{D}_{\dot{t}_i} H + \mathbf{D}_{\dot{t}_{i+1}} H.$$

Substituting this into Equation (5.30) and replacing i by $i - 1$ yields

$$\mathbf{D}_{\dot{t}_i} H(E_{i-1}(t)) = (\mu_{i-1}(t) + 2 \cos \frac{2\pi}{n}) \mathbf{D}_{\dot{u}_{i-1}} G_{i-1}(t) + \nu_{i-1}(t) \mathbf{D}_{\dot{v}_{i-1}} G_{i-1}(t). \quad (5.37)$$

Setting $t = 1$, and using

$$\mathbf{D}_{\dot{t}_i} H(p_i) = \mathbf{D}_{\dot{u}_i} G_i(0) = \mathbf{D}_{\dot{r}_{i-1}} G_{i-1}(1),$$

it follows that

$$\mathbf{D}_{\dot{r}_{i-1}} G_{i-1}(1) = (\mu_{i-1}(1) + 2 \cos \frac{2\pi}{n}) \mathbf{D}_{\dot{u}_{i-1}} G_{i-1}(1) + \nu_{i-1}(1) \mathbf{D}_{\dot{v}_{i-1}} G_{i-1}(1).$$

Hence using Equation (5.36)

$$\mu_i(1) = -\alpha_{i+1} - 2 \cos \frac{2\pi}{n}, \quad \nu_i(1) = \beta_{i+1}, \quad 1 = 1, \dots, n.$$

Thus far, the values of μ_i and ν_i , $i = 1, \dots, n$, have been constrained at the endpoints 0 and 1. The final set of constraints involves the derivatives μ'_i and ν'_i at the endpoints. These constraints are found by requiring that μ_i, ν_i , $i = 1, \dots, n$ must be constructed so that the resulting S-patch is twist compatible, i.e.

$$\mathbf{D}_{-\dot{t}_{i-1}, \dot{t}_i}^2 H(p_i) = \mathbf{D}_{\dot{t}_i, -\dot{t}_{i-1}}^2 H(p_i). \quad (5.38)$$

The left hand side of Equation (5.38) is expanded by differentiating Equation (5.30) with respect to \dot{t}_i and evaluating at $t = 0$, resulting in

$$\begin{aligned} \mathbf{D}_{-\dot{t}_{i-1}, \dot{t}_i}^2 H(E_i(0)) &= \mu'_i(0) \mathbf{D}_{\dot{u}_i} G_i(0) - \mu_i(0) \mathbf{D}_{\dot{u}_i, \dot{u}_i}^2 G_i(0) \\ &+ \nu'_i(0) \mathbf{D}_{\dot{v}_i} G_i(0) - \nu_i(0) \mathbf{D}_{\dot{v}_i, \dot{u}_i}^2 G_i(0). \end{aligned} \quad (5.39)$$

The right hand side of Equation (5.38) is expanded by differentiating Equation (5.37) with respect to $-\dot{t}_{i-1}$ and evaluating at $t = 1$, resulting in

$$\begin{aligned}
\mathbf{D}_{\dot{t}_i, -\dot{t}_{i-1}}^2 H(E_{i-1}(1)) &= -\mu'_{i-1}(1) \mathbf{D}_{\dot{u}_{i-1}} G_{i-1}(1) \\
&+ (\mu_{i-1}(1) + 2 \cos \frac{2\pi}{n}) \mathbf{D}_{\dot{u}_{i-1}, \dot{u}_{i-1}}^2 G_{i-1}(1) \\
&- \nu'_{i-1}(1) \mathbf{D}_{\dot{v}_{i-1}} G_{i-1}(1) \\
&+ \nu_{i-1}(1) \mathbf{D}_{\dot{v}_{i-1}, \dot{u}_{i-1}}^2 G_{i-1}(1). \tag{5.40}
\end{aligned}$$

From Equations (5.35) and (5.36), Conditions (5.31), (5.32), and (5.33), and by linearity of differentiation, the following identities hold:

$$\begin{aligned}
\mathbf{D}_{\dot{v}_i} G_i(0) &= \frac{1}{\beta_i} \mathbf{D}_{\dot{s}_i} G_i(0) - \frac{\alpha_i}{\beta_i} \mathbf{D}_{\dot{u}_i} G_i(0), \\
\mathbf{D}_{\dot{v}_i, \dot{u}_i}^2 G_i(0) &= \frac{1}{\beta_i} \mathbf{D}_{\dot{s}_i, \dot{u}_i}^2 G_i(0) - \frac{\alpha_i}{\beta_i} \mathbf{D}_{\dot{u}_i, \dot{u}_i}^2 G_i(0), \\
\mathbf{D}_{\dot{v}_{i-1}} G_{i-1}(1) &= \frac{1}{\beta_i} \mathbf{D}_{\dot{r}_{i-1}} G_{i-1}(1) + \frac{\alpha_i}{\beta_i} \mathbf{D}_{\dot{u}_{i-1}} G_{i-1}(1), \\
&= \frac{1}{\beta_i} \mathbf{D}_{\dot{u}_i} G_i(0) - \frac{\alpha_i}{\beta_i} \mathbf{D}_{\dot{s}_i} G_i(0), \\
\mathbf{D}_{\dot{v}_{i-1}, \dot{u}_{i-1}}^2 G_{i-1}(1) &= \frac{1}{\beta_i} \mathbf{D}_{\dot{r}_{i-1}, \dot{u}_{i-1}}^2 G_{i-1}(1) + \frac{\alpha_i}{\beta_i} \mathbf{D}_{\dot{u}_{i-1}, \dot{u}_{i-1}}^2 G_{i-1}(1), \\
&= -\frac{1}{\beta_i} \mathbf{D}_{\dot{u}_i, \dot{s}_i}^2 G_i(0) + \frac{\alpha_i}{\beta_i} \mathbf{D}_{\dot{s}_i, \dot{s}_i}^2 G_i(0).
\end{aligned}$$

Substituting these identities into Equations (5.39) and (5.40) shows

$$\begin{aligned}
&\mu'_i(0) \mathbf{D}_{\dot{u}_i} G_i(0) - \mu_i(0) \mathbf{D}_{\dot{u}_i, \dot{u}_i}^2 G_i(0) + \\
&\nu'_i(0) \left(\frac{1}{\beta_i} \mathbf{D}_{\dot{s}_i} G_i(0) - \frac{\alpha_i}{\beta_i} \mathbf{D}_{\dot{u}_i} G_i(0) \right) - \\
&\nu_i(0) \left(\frac{1}{\beta_i} \mathbf{D}_{\dot{s}_i, \dot{u}_i}^2 G_i(0) - \frac{\alpha_i}{\beta_i} \mathbf{D}_{\dot{u}_i, \dot{u}_i}^2 G_i(0) \right) \\
&= \\
&\mu'_{i-1}(1) \mathbf{D}_{\dot{s}_i} G_i(0) + (\mu_{i-1}(1) + 2 \cos \frac{2\pi}{n}) \mathbf{D}_{\dot{s}_i, \dot{s}_i}^2 G_i(0) -
\end{aligned}$$

$$\begin{aligned} & \nu'_{i-1}(1) \left(\frac{1}{\beta_i} \mathbf{D}_{\dot{u}_i} G_i(0) - \frac{\alpha_i}{\beta_i} \mathbf{D}_{\dot{s}_i} G_i(0) \right) + \\ & \nu_{i-1}(1) \left(-\frac{1}{\beta_i} \mathbf{D}_{\dot{u}_i, \dot{s}_i}^2 G_i(0) + \frac{\alpha_i}{\beta_i} \mathbf{D}_{\dot{s}_i, \dot{s}_i}^2 G_i(0) \right). \end{aligned}$$

This expression is rearranged to get

$$\begin{aligned} & \left[\mu'_i(0) - \frac{\alpha_i}{\beta_i} \nu'_i(0) + \frac{1}{\beta_i} \nu'_{i-1}(1) \right] \mathbf{D}_{\dot{u}_i} G_i(0) + \\ & \left[\frac{1}{\beta_i} \nu'_i(0) - \mu'_{i-1}(1) - \frac{\alpha_i}{\beta_i} \nu'_{i-1}(1) \right] \mathbf{D}_{\dot{s}_i} G_i(0) + \\ & \left[-\mu_i(0) + \frac{\alpha_i}{\beta_i} \nu_i(0) \right] \mathbf{D}_{\dot{u}_i, \dot{u}_i}^2 G_i(0) + \\ & \left[-\frac{1}{\beta_i} \nu_i(0) + \frac{1}{\beta_i} \nu_{i-1}(1) \right] \mathbf{D}_{\dot{s}_i, \dot{u}_i}^2 G_i(0) + \\ & \left[-\mu_{i-1}(1) - 2 \cos \frac{2\pi}{n} - \frac{\alpha_i}{\beta_i} \nu_{i-1}(1) \right] \mathbf{D}_{\dot{s}_i, \dot{s}_i}^2 G_i(0) = 0. \end{aligned}$$

This equality is satisfied, in general, if and only if

$$\beta_i \mu'_i(0) = \alpha_i \nu'_i(0) - \nu'_{i-1}(1), \quad (5.41)$$

$$\nu'_i(0) = \beta_i \mu'_{i-1}(1) + \alpha_i \nu'_{i-1}(1). \quad (5.42)$$

To summarize, the constraints that μ_i and $\nu_i, i = 1, \dots, n$, must satisfy are

$$\begin{aligned} \mu_i(0) &= \alpha_i, & \nu_i(0) &= \beta_i, \\ \beta_i \mu'_i(0) &= \alpha_i \nu'_i(0) - \nu'_{i-1}(1), & \nu'_i(0) &= \beta_i \mu'_{i-1}(1) + \alpha_i \nu'_{i-1}(1), \\ \mu_i(1) &= -\alpha_{i+1} - 2 \cos \frac{2\pi}{n}, & \nu_i(1) &= \beta_{i+1}. \end{aligned}$$

The next step is to solve for the unknowns $\mu'_i(0), \nu'_i(0), \mu'_i(1)$, and $\nu'_i(1), i = 1, \dots, n$. This is done by setting up a linear system in the polynomial coefficients of μ_i and ν_i .

Quadratic Solution

The constraints can be satisfied if μ_i and ν_i are assumed to be quadratic. Unfortunately, the quadratic solution has rather peculiar asymmetric behavior that renders it unusable in general. The derivation of the quadratic solution is included here for completeness. This section can safely be skipped without any loss of continuity.

Let μ_i and ν_i be quadratic polynomials with Bernstein coefficients $\mu_{i,0}, \mu_{i,1}, \mu_{i,2}$ and $\nu_{i,0}, \nu_{i,1}, \nu_{i,2}$ respectively. Using a basic property of Bernstein polynomials it follows that

$$\begin{aligned}\mu'_i(0) &= 2(\mu_{i,1} - \mu_{i,0}), & \nu'_i(0) &= 2(\nu_{i,1} - \nu_{i,0}), \\ \mu'_i(1) &= 2(\mu_{i,2} - \mu_{i,1}), & \nu'_i(1) &= 2(\nu_{i,2} - \nu_{i,1}).\end{aligned}$$

Also note that

$$\begin{aligned}\mu_{i,0} &= \alpha_i, & \nu_{i,0} &= \beta_i, \\ \mu_{i,2} &= -\alpha_{i+1} - 2\cos\frac{2\pi}{n}, & \nu_{i,2} &= \beta_{i+1}.\end{aligned}$$

Using these facts, Constraints (5.41) and (5.42) are rewritten

$$\begin{aligned}\beta_i\mu_{i,1} &= \nu_{i-1,1} + \alpha_i\nu_{i,1} - \beta_i, \\ \beta_i\mu_{i-1,1} &= \beta_i(1 - 2\cos\frac{2\pi}{n}) - \alpha_i\nu_{i-1,1} - \nu_{i,1},\end{aligned}$$

where $i = 1, \dots, n$.

These constraints form a system of $2n$ equations in $2n$ unknowns. This system may be reduced to $n \times n$ by dividing the right hand side by β_i , substituting i for $i - 1$ in the second constraint and equating to get

$$\beta_{i+1}\nu_{i-1,1} + (\alpha_i\beta_{i+1} + \beta_i\alpha_{i+1})\nu_{i,1} + \beta_i\nu_{i+1,1} = \beta_i\beta_{i+1}(2 - 2\cos\frac{2\pi}{n}).$$

Once this system has been solved for $\nu_{1,1}, \dots, \nu_{n,1}$, $\mu_{1,1}, \dots, \mu_{n,1}$ may be found by substitution into either constraint.

Cubic Solution

Suppose μ_i and ν_i are cubic polynomials with Bernstein ordinates $\mu_{i,0}, \mu_{i,1}, \mu_{i,2}, \mu_{i,3}$ and $\nu_{i,0}, \nu_{i,1}, \nu_{i,2}, \nu_{i,3}$ respectively. It follows that

$$\begin{aligned}\mu'_i(0) &= 3(\mu_{i,1} - \mu_{i,0}), & \nu'_i(0) &= 3(\nu_{i,1} - \nu_{i,0}), \\ \mu'_i(1) &= 3(\mu_{i,3} - \mu_{i,2}), & \nu'_i(1) &= 3(\nu_{i,3} - \nu_{i,2}).\end{aligned}$$

Also it is known that

$$\begin{aligned}\mu_{i,0} &= \alpha_i, & \nu_{i,0} &= \beta_i, \\ \mu_{i,3} &= -\alpha_{i+1} - 2 \cos \frac{2\pi}{n}, & \nu_{i,3} &= \beta_{i+1}.\end{aligned}$$

Using these facts, Constraints (5.41) and (5.42) become

$$\beta_i \mu_{i,1} = \nu_{i-1,2} + \alpha_i \nu_{i,1} - \beta_i, \quad (5.43)$$

$$\beta_i \mu_{i-1,2} = \beta_i (1 - 2 \cos \frac{2\pi}{n}) - \alpha_i \nu_{i-1,2} - \nu_{i,1}. \quad (5.44)$$

These constraints form a system of $2n$ equations in $4n$ unknowns. This leaves $2n$ degrees of freedom, so many solutions are possible. The system may be simplified to n equations in $3n$ unknowns by isolating $\nu_{i-1,2}$ in both equations, then equating. This gives

$$(\alpha_i \alpha_i - 1) \nu_{i,1} = \alpha_i \beta_i \mu_{i,1} + \beta_i \mu_{i-1,2} + \beta_i (\alpha_i - 1 + 2 \cos \frac{2\pi}{n}) \quad (5.45)$$

A system of $n \times n$ equations is obtained if $\mu_i, i = 1, \dots, n$, are assumed to be linear.

Under this assumption, it can be deduced from degree elevation that

$$\begin{aligned}\mu_{i,1} &= \frac{2\alpha_i - \alpha_{i+1} - 2 \cos \frac{2\pi}{n}}{3}, \\ \mu_{i,2} &= \frac{\alpha_i - 2\alpha_{i+1} - 4 \cos \frac{2\pi}{n}}{3}.\end{aligned}$$

Substituting these values into Equation (5.45) shows that $\nu_{i,1}$ may be found directly.

Values of $\nu_{i,2}$ may be found by substituting into either constraint (5.43) or (5.44).

Using the known values $\alpha_i = \cos \frac{2\pi}{k_i}$ and $\beta_i = \sin \frac{2\pi}{k_i}$, the Bernstein coefficients of μ_i and ν_i can be written

$$\mu_{i,0} = \cos \frac{2\pi}{k_i}, \quad (5.46)$$

$$\mu_{i,1} = \frac{2 \cos \frac{2\pi}{k_i} - \cos \frac{2\pi}{k_{i+1}} - 2 \cos \frac{2\pi}{n}}{3}, \quad (5.47)$$

$$\mu_{i,2} = \frac{\cos \frac{2\pi}{k_i} - 2 \cos \frac{2\pi}{k_{i+1}} - 4 \cos \frac{2\pi}{n}}{3}, \quad (5.48)$$

$$\mu_{i,3} = -\cos \frac{2\pi}{k_{i+1}} - 2 \cos \frac{2\pi}{n}, \quad (5.49)$$

and

$$\nu_{i,0} = \sin \frac{2\pi}{k_i}, \quad (5.50)$$

$$\nu_{i,1} = \frac{3 - \cos \frac{2\pi}{k_{i-1}} - 2 \cos \frac{2\pi}{n} - \cos \frac{2\pi}{k_i} (1 + 2 \cos \frac{2\pi}{k_i} - \cos \frac{2\pi}{k_{i+1}} - 2 \cos \frac{2\pi}{n})}{3 \sin \frac{2\pi}{k_i}}, \quad (5.51)$$

$$\nu_{i,2} = \frac{3 - \cos \frac{2\pi}{k_{i+2}} - 2 \cos \frac{2\pi}{n} - \cos \frac{2\pi}{k_{i+1}} (1 - \cos \frac{2\pi}{k_i} + 2 \cos \frac{2\pi}{k_{i+1}} - 2 \cos \frac{2\pi}{n})}{3 \sin \frac{2\pi}{k_{i+1}}}, \quad (5.52)$$

$$\nu_{i,3} = \sin \frac{2\pi}{k_{i+1}}. \quad (5.53)$$

Note that for an edge $(\mathbf{v}_i, \mathbf{v}_{i+1}) \in \mathbf{E}$ only the orders k_{i-1}, k_i, k_{i+1} , and k_{i+2} of the vertices $\mathbf{v}_{i-1}, \mathbf{v}_i, \mathbf{v}_{i+1}$, and \mathbf{v}_{i+2} respectively are involved in computing μ_i and ν_i . Since μ_i given above is actually linear, it may be written equivalently as

$$\mu_i(u) = \cos \frac{2\pi}{k_i} (1 - u) + [-\cos \frac{2\pi}{k_{i+1}} - 2 \cos \frac{2\pi}{n}] u.$$

Once the functions μ_i and ν_i have been determined, the differential $\mathbf{D}H(E_i(t)), i = 1, \dots, n$ is determined for the boundary of the domain polygon P by Equation (5.30). In the next section, this information is used to compute the S -patch control points that characterize its boundary differential.

5.5.2 Boundary Panels

The S -patch control points that characterize the boundary differential are collectively referred to as *boundary panels*. These are the S -patch panels along the boundary of H . In this section, the control points of the S -patch boundary panels are computed from $H(E_i(t))$ and $\mathbf{D}H(E_i(t)), i = 1, \dots, n$.

In order to compute the S -patch control points of H , some discussion of the depth of H is in order. Recall that the depth of an S -patch is equal to the degree of its defining Bézier simplex (see Section 4.1). From Equation (5.30):

$$\mathbf{D}_{-i_{i-1}} H(E_i(t)) = \mu_i(t) \mathbf{D}_{\dot{u}_i} G_i(t) + \nu_i(t) \mathbf{D}_{\dot{v}_i} G_i(t),$$

it is clear that $\text{degree}(\mathbf{D}_{-i_{i-1}} H(E_i(t)))$ is equal to the greater of $\text{degree}(\mu_i) + \text{degree}(\mathbf{D}_{i_i} G_i)$ and $\text{degree}(\nu_i) + \text{degree}(\mathbf{D}_{i_i} G_i)$. Therefore it follows that

$$\text{depth}(H) \geq \max(\text{degree}(\mu_i) + \text{degree}(G_i), \text{degree}(\nu_i) + \text{degree}(\mathbf{D}_{i_i} G_i) + 1),$$

In the general case, it has been established in the previous sections that

$$\text{degree}(G_i) = 5, \quad \text{degree}(\mathbf{D}_{i_i} G_i) = 3, \quad \text{degree}(\mu_i) = 1, \quad \text{degree}(\nu_i) = 3.$$

Therefore

$$\text{depth}(H) \geq 7.$$

Many special cases exist for which $\text{depth}(H) < 7$, these are discussed in the Chapter 6.

Since the boundary behavior of H is strictly polynomial (as opposed to rational), the boundary panels are all planar affine n -gons. Therefore, once three points of each boundary panel are known, the remaining points can be uniquely determined. The procedure for finding the boundary panels is as follows: first, the control points of the boundary curves of H are found, yielding at least two points for each boundary panel; second, $\mathbf{D}_{-i_{i-1}} H(E_i(t))$ is used to compute a third point for each panel; finally, the remaining $n - 3$ points for each boundary panel are found by taking affine combinations of the first three points found.

Since the boundary of H is found as

$$H(E_i(t)) = G_i(t), \quad i = 1, \dots, n,$$

the boundary control points of H are computed by degree raising $G_i(t)$. The degree 5 curve $G_i(t)$ is degree raised to obtain the control points of the degree 7 boundary curve $H(E_i(t))$ by the following formulae

$$\begin{aligned} \mathbf{h}_{7\bar{e}_i} &= \mathbf{c}_{i,0}, \\ \mathbf{h}_{6\bar{e}_i + \bar{e}_{i+1}} &= \frac{5}{7}\mathbf{c}_{i,1} + \frac{2}{7}\mathbf{c}_{i,0}, \\ \mathbf{h}_{5\bar{e}_i + 2\bar{e}_{i+1}} &= \frac{10}{21}\mathbf{c}_{i,2} + \frac{10}{21}\mathbf{c}_{i,1} + \frac{1}{21}\mathbf{c}_{i,0}, \end{aligned}$$

$$\begin{aligned}
\mathbf{h}_{4\bar{e}_i+3\bar{e}_{i+1}} &= \frac{10}{35}\mathbf{c}_{i,3} + \frac{20}{35}\mathbf{c}_{i,2} + \frac{5}{35}\mathbf{c}_{i,1}, \\
\mathbf{h}_{3\bar{e}_i+4\bar{e}_{i+1}} &= \frac{5}{35}\mathbf{c}_{i,4} + \frac{20}{35}\mathbf{c}_{i,3} + \frac{10}{35}\mathbf{c}_{i,2}, \\
\mathbf{h}_{2\bar{e}_i+5\bar{e}_{i+1}} &= \frac{1}{21}\mathbf{c}_{i,5} + \frac{10}{21}\mathbf{c}_{i,4} + \frac{10}{21}\mathbf{c}_{i,3}, \\
\mathbf{h}_{\bar{e}_i+6\bar{e}_{i+1}} &= \frac{2}{7}\mathbf{c}_{i,5} + \frac{5}{7}\mathbf{c}_{i,4},
\end{aligned}$$

where $\mathbf{c}_{i,0}, \dots, \mathbf{c}_{i,5}$ are the Bézier control points of G_i from Section 5.4. This step will generate at least two points for each boundary panel.

When $\mathbf{D}H(E_i(t))$ has polynomial behavior, a third point on each panel can be found by writing $\mathbf{D}_{-i_{i-1}}H(E_i(t))$ in terms of its S -patch control points

$$\mathbf{D}_{-i_{i-1}}H(E_i(t)) = 7 \sum_{k=0}^6 (\mathbf{h}_{[\bar{e}_{i-1}+(6-k)\bar{e}_i+k\bar{e}_{i+1}]} - \mathbf{h}_{[(7-k)\bar{e}_i+k\bar{e}_{i+1}]})B_k^6(t).$$

This is rearranged to get

$$\sum_{k=0}^6 \mathbf{h}_{[\bar{e}_{i-1}+(6-k)\bar{e}_i+k\bar{e}_{i+1}]}B_k^6(t) = \sum_{k=0}^6 \mathbf{h}_{[(7-k)\bar{e}_i+k\bar{e}_{i+1}]}B_k^6(t) + \frac{1}{7}\mathbf{D}_{-i_{i-1}}H(E_i(t)). \quad (5.54)$$

Substituting the definitions of $\mathbf{D}_{-i_{i-1}}H(E_i(t))$, $G_i(t)$, $\mathbf{D}_v G_i(t)$, $\mu_i(t)$, and $\nu_i(t)$ into Equation (5.54) results in

$$\begin{aligned}
\mathbf{h}_{\bar{e}_{i-1}+5\bar{e}_i+\bar{e}_{i+1}} &= \frac{12-5\mu_{i,0}-5\mu_{i,3}}{42}\mathbf{c}_{i,0} + \frac{30-15\mu_{i,0}+5\mu_{i,3}}{42}\mathbf{c}_{i,1} + \frac{10\mu_{i,0}}{21}\mathbf{c}_{i,2} \\
&\quad + \frac{\nu_{i,1}}{14}\dot{\mathbf{c}}_{i,0} + \frac{\nu_{i,0}}{14}\dot{\mathbf{c}}_{i,1}, \\
\mathbf{h}_{\bar{e}_{i-1}+4\bar{e}_i+2\bar{e}_{i+1}} &= \frac{1-\mu_{i,3}}{21}\mathbf{c}_{i,0} + \frac{10-4\mu_{i,0}-3\mu_{i,3}}{21}\mathbf{c}_{i,1} + \frac{10-2\mu_{i,0}+4\mu_{i,3}}{21}\mathbf{c}_{i,2} + \frac{2\mu_{i,0}}{7}\mathbf{c}_{i,3} \\
&\quad + \frac{\nu_{i,2}}{35}\dot{\mathbf{c}}_{i,0} + \frac{3\nu_{i,1}}{35}\dot{\mathbf{c}}_{i,1} + \frac{\nu_{i,0}}{35}\dot{\mathbf{c}}_{i,2}, \\
\mathbf{h}_{\bar{e}_{i-1}+3\bar{e}_i+3\bar{e}_{i+1}} &= \frac{1-\mu_{i,3}}{7}\mathbf{c}_{i,1} + \frac{8-3\mu_{i,0}-\mu_{i,3}}{14}\mathbf{c}_{i,2} + \frac{4+\mu_{i,0}+3\mu_{i,3}}{14}\mathbf{c}_{i,3} + \frac{\mu_{i,0}}{7}\mathbf{c}_{i,4} \\
&\quad + \frac{\nu_{i,3}}{140}\dot{\mathbf{c}}_{i,0} + \frac{9\nu_{i,2}}{140}\dot{\mathbf{c}}_{i,1} + \frac{9\nu_{i,1}}{140}\dot{\mathbf{c}}_{i,2} + \frac{\nu_{i,0}}{140}\dot{\mathbf{c}}_{i,3}, \\
\mathbf{h}_{\bar{e}_{i-1}+2\bar{e}_i+4\bar{e}_{i+1}} &= \frac{2-2\mu_{i,3}}{7}\mathbf{c}_{i,2} + \frac{12-4\mu_{i,0}+2\mu_{i,3}}{21}\mathbf{c}_{i,3} + \frac{3+3\mu_{i,0}+4\mu_{i,3}}{21}\mathbf{c}_{i,4} + \frac{\mu_{i,0}}{21}\mathbf{c}_{i,5}
\end{aligned}$$

$$\begin{aligned}
& + \frac{\nu_{i,3}}{35} \dot{\mathbf{c}}_{i,1} + \frac{3\nu_{i,2}}{35} \dot{\mathbf{c}}_{i,2} + \frac{\nu_{i,1}}{35} \dot{\mathbf{c}}_{i,3}, \\
\mathbf{h}_{\vec{e}_{i-1} + \vec{e}_i + 5\vec{e}_{i+1}} & = \frac{10-10\mu_{i,3}}{21} \mathbf{c}_{i,3} + \frac{20-5\mu_{i,0}+15\mu_{i,3}}{42} \mathbf{c}_{i,4} + \frac{2+5\mu_{i,0}+5\mu_{i,3}}{42} \mathbf{c}_{i,5} \\
& + \frac{\nu_{i,3}}{14} \dot{\mathbf{c}}_{i,2} + \frac{\nu_{i,2}}{14} \dot{\mathbf{c}}_{i,3},
\end{aligned}$$

where $\dot{\mathbf{c}}_{i,0}, \dots, \dot{\mathbf{c}}_{i,3}$ are the Bézier control vectors of $\mathbf{D}_{\dot{v}_i} G_i(t)$ found by Equations (5.24) through (5.27).

The boundary panels are completed by taking appropriate affine combination of the three points found thus far. This is done with the following CompletePanels algorithm (with $d = 7$ below):

for $i = 2$ *to* $n - 2$

$$\begin{aligned}
r & \leftarrow \frac{1 - \cos \frac{2\pi i}{n} - \tan \frac{\pi}{n} \sin \frac{2\pi i}{n}}{2(1 - \cos \frac{2\pi}{n})} \\
s & \leftarrow \frac{\cos \frac{2\pi i}{n} - \cos \frac{2\pi}{n}}{1 - \cos \frac{2\pi}{n}} \\
t & \leftarrow \frac{1 - \cos \frac{2\pi i}{n} + \tan \frac{\pi}{n} \sin \frac{2\pi i}{n}}{2(1 - \cos \frac{2\pi}{n})}
\end{aligned}$$

for $j = 1$ *to* n

for $k = 1$ *to* $d - 1$

$$\vec{i} \leftarrow (d - k)\vec{e}_j + (k - 1)\vec{e}_{j+1}$$

$$\mathbf{h}_{\vec{i} + \vec{e}_{i+j}} \leftarrow r\mathbf{h}_{\vec{i} + \vec{e}_{j-1}} + s\mathbf{h}_{\vec{i} + \vec{e}_j} + t\mathbf{h}_{\vec{i} + \vec{e}_{j+1}}$$

end for

end for

end for

The scalars r , s , and t are the barycentric coordinates with respect to triangle $p_n p_1 p_2$ of the k^{th} vertex of an affine n -gon p_1, \dots, p_n . An S -patch with completed boundary panels

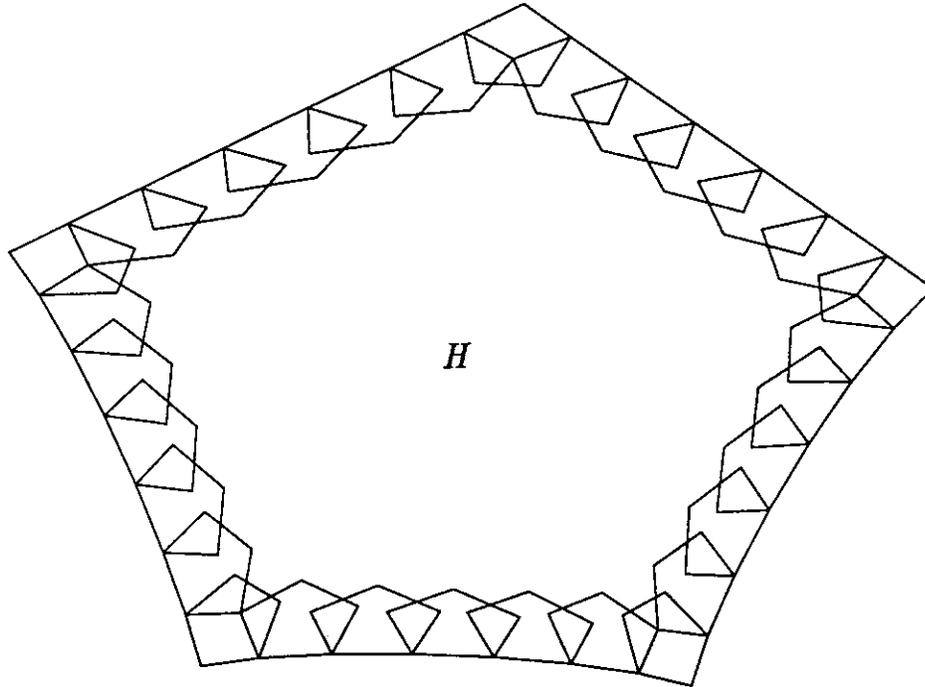


Figure 5.5: The boundary panels of S -patch H

is shown in Figure 5.5.

5.5.3 Interior Points

In this section, a procedure for determining the interior control points of an S -patch H is given. This particular procedure has no special significance, other than it is relatively simple, the resulting surfaces are aesthetically well behaved, and it generalizes regular B-splines. This last property is given more attention in Chapter 6. The basic procedure is as follows: a depth 4 S -patch is constructed that is approximately equivalent to the depth 7 S -patch whose interior control points are unknown. This *auxiliary patch* is then depth elevated to depth 7 using the Bézier simplex degree elevation algorithm of Section 2.2.4.

In what follows, let $P = \{p_1, \dots, p_n\}$ be a regular n -gon in a 2 dimensional affine

space; let $f = \{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ be the control mesh face associated with the S-patch H whose interior points are unknown; and let A be a depth 4 S-patch with control points $\mathbf{a}_{\vec{i}}$. The idea is to describe the control net of A on the face f , where f is the depth 1 S-patch with vertices $\mathbf{v}_1, \dots, \mathbf{v}_n$. The boundary panels of A are then modified, and A is subsequently depth elevated.

Step 1. Compute the initial control points of A as follows:

$$\mathbf{a}_{\vec{i}} = \ell_1(q_{\vec{i}})\mathbf{v}_1 + \dots + \ell_n(q_{\vec{i}})\mathbf{v}_n,$$

where $\vec{i} = (i_1, \dots, i_n)$, $|\vec{i}| = 4$, and where

$$q_{\vec{i}} = \frac{i_1 p_1 + \dots + i_n p_n}{n}.$$

The functions ℓ_1, \dots, ℓ_n come from the S-patch embedding given in Section 4.1.

Step 2. Three control points from each boundary panel of A are modified as follows:

$$\begin{aligned} \mathbf{a}_{4\vec{e}_i} &= \mathbf{c}_{0,i}, \\ \mathbf{a}_{3\vec{e}_i + \vec{e}_{i+1}} &= \frac{5}{4}\mathbf{c}_{1,i} - \frac{1}{4}\mathbf{c}_{0,i}, \\ \mathbf{a}_{2\vec{e}_i + 2\vec{e}_{i+1}} &= \frac{1}{12}\mathbf{c}_{0,i} - \frac{5}{12}\mathbf{c}_{1,i} + \frac{5}{6}\mathbf{c}_{2,i} + \frac{5}{6}\mathbf{c}_{3,i} - \frac{5}{12}\mathbf{c}_{4,i} + \frac{1}{12}\mathbf{c}_{5,i}, \\ \mathbf{a}_{\vec{e}_i + 3\vec{e}_{i+1}} &= \frac{5}{4}\mathbf{c}_{4,i} - \frac{1}{4}\mathbf{c}_{5,i}, \\ \mathbf{a}_{\vec{e}_{i-1} + 2\vec{e}_i + \vec{e}_{i+1}} &= \frac{10\mu_{i,0} - 5\mu_{i,3} - 3}{12}\mathbf{c}_{i,0} + \frac{15 - 30\mu_{i,0} + 5\mu_{i,3}}{12}\mathbf{c}_{i,1} + \frac{10\mu_{i,0}}{6}\mathbf{c}_{i,2} \\ &\quad + \frac{\nu_{i,1} - \nu_{i,0}}{4}\dot{\mathbf{c}}_{i,0} + \frac{\nu_{i,0}}{4}\dot{\mathbf{c}}_{i,1}, \\ \mathbf{a}_{\vec{e}_{i-1} + \vec{e}_i + 2\vec{e}_{i+1}} &= \frac{10 - 10\mu_{i,3}}{6}\mathbf{c}_{i,3} + \frac{30\mu_{i,3} - 5\mu_{i,0} - 10}{12}\mathbf{c}_{i,4} + \frac{2 + 5\mu_{i,0} - 10\mu_{i,3}}{12}\mathbf{c}_{i,5} \\ &\quad + \frac{\nu_{i,3}}{4}\dot{\mathbf{c}}_{i,2} + \frac{\nu_{i,2} - \nu_{i,3}}{4}\dot{\mathbf{c}}_{i,3}, \end{aligned}$$

where $i = 1, \dots, n$. These modifications are intended to endow A with a boundary behavior approximately the same as the depth 7 S-patch H constructed in the previous sections.

Step 3. Run the CompletePanels procedure given in Section 5.5.3 on the control net of A . After this procedure is run, the positions, tangents, and mixed partials (in

boundary curve directions) of A at patch corners will be identical to those of the S-patch H . The S-patches A and H will differ slightly in position and tangent plane along their boundaries. However unlike H , the control net of S-patch A is completely defined.

Step 4. Elevate the control net of A to depth 7 and equate the unknown interior control points of H with those of A . The S-patch depth elevation procedure is equivalent to the Bézier simplex degree raising formula given in Section 2.2.4.

The procedure for determining interior control points outlined in Steps 1 through 4 has a few unnecessary operations that an optimized implementation might want to avoid. In Step 1, there is no need to compute points belonging to boundary panels since, in Steps 2 and 3, the boundary panels of A are re-computed. Also, in Step 3 some control points are found redundantly by the CompletePanels procedure. These details are ignored here in the interest of simplicity.

The construction of H is completed once the interior points have been determined. Computing the boundary panels of H , then determining the interior points is a logical progression when presenting this construction. However, it may be more space efficient first to determine the interior points, then to compute the boundary panels. The interior control point construction just presented does not depend on the boundary panel construction. So these two stages could be run in either order (or in parallel).

5.6 Summary

In this chapter, the construction for the generalized B-spline scheme based on S-patches was presented. This construction proceeded in three phases. In the first phase, a collection of Bézier control points was constructed corresponding to control mesh vertices. These Bézier control points formed a “vertex map” that characterized up to second order, the behavior of the spline surfaces at points shared by several patches. In the second phase, “edge maps” were constructed whose endpoint behavior matched a pair of vertex maps from the previous step. The positions and derivatives of these endpoint vertex maps were blended at all other points of an edge map. Finally, in the third phase, an

S-patch was constructed to interpolate the boundary curves and transversal vector fields prescribed by the surrounding edge maps.

Several examples of control meshes and the resulting S-patch surfaces are shown in Figures 5.6 through 5.10. In these figures each S-patch has been colored to distinguish it from neighboring patches.

In the next chapter, it is demonstrated that the S-patches determined by the generalized B-spline scheme may, in certain special cases, be lower than depth 7. These results are key to showing that this general construction generalizes regular B-splines.

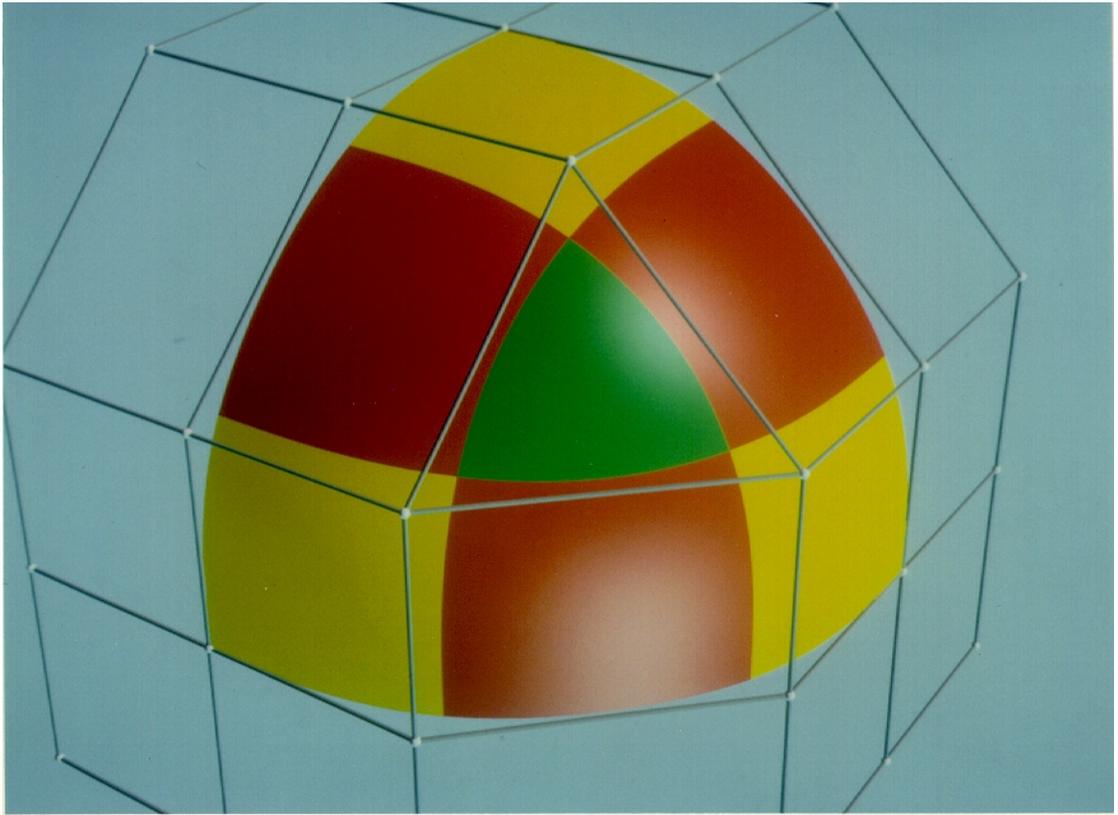


Figure 5.6: The classic "suitcase corner" problem



Figure 5.7: A 5-sided patch in an otherwise rectangular control mesh

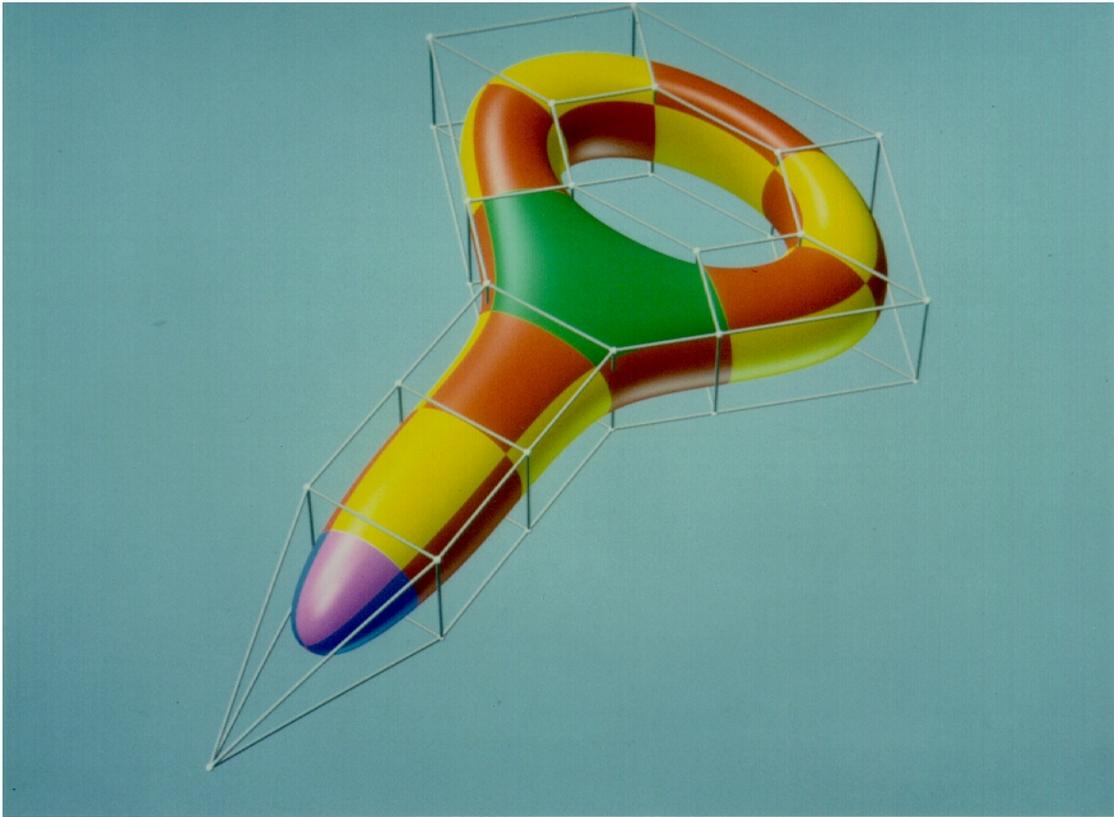


Figure 5.8: A closed surface with a handle

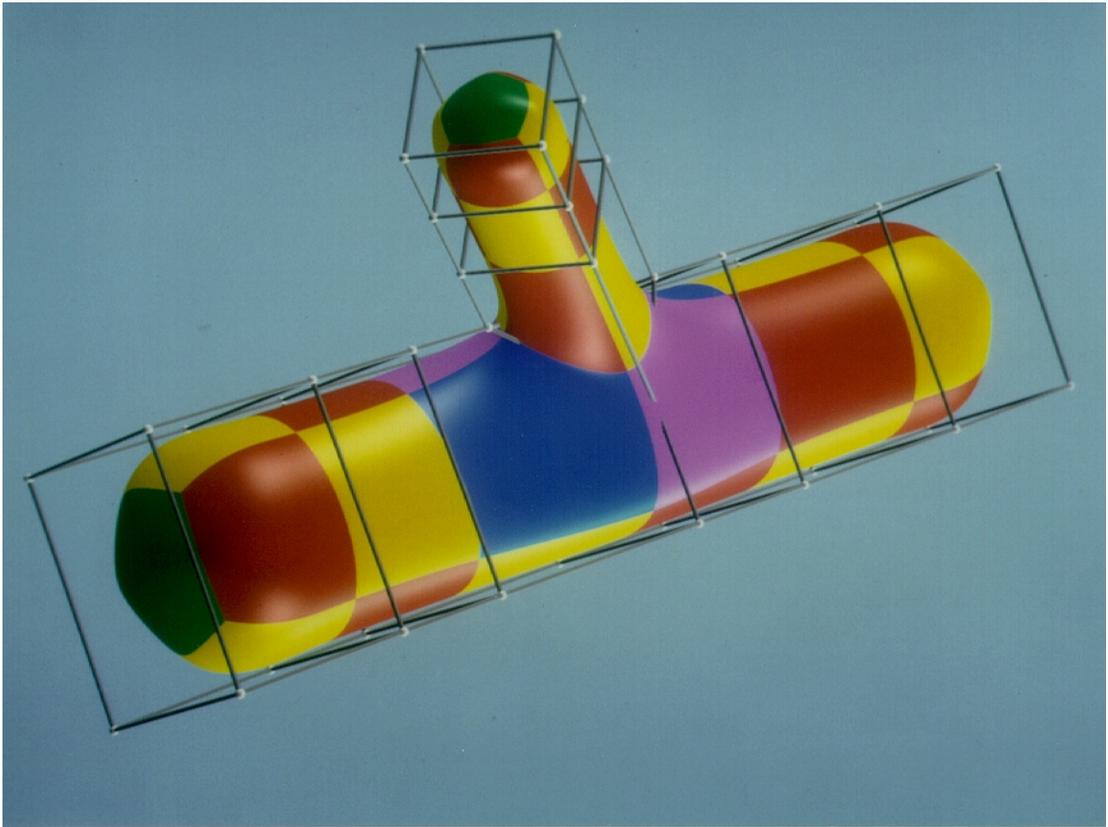


Figure 5.9: A closed branching surface

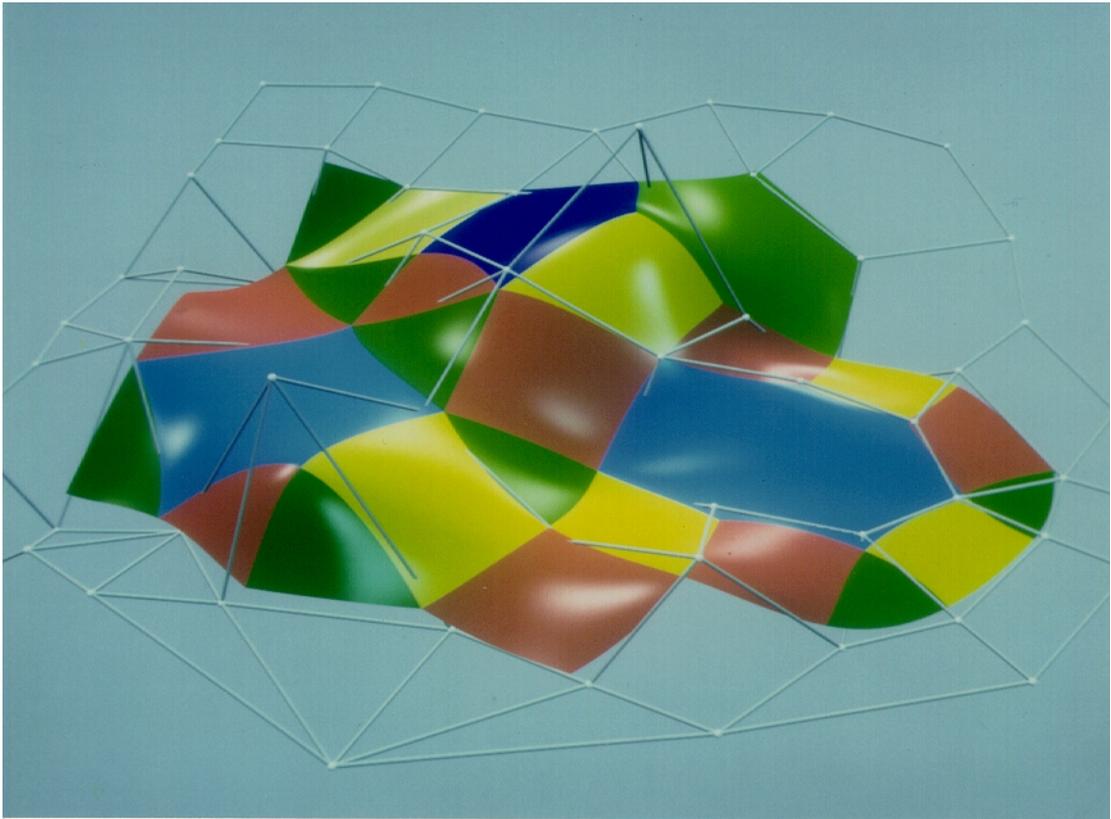


Figure 5.10: A surface with an irregular control net

Chapter 6

Specializations

The primary goal of this work has been to create a general purpose geometric construction for generating smooth spline surfaces. The resulting surfaces mimic the behavior of B-spline surfaces, yet are not limited by the constraints imposed on the form of the control mesh. Simply behaving like a regular B-spline has not been the only goal. In cases where the control mesh satisfies the constraints of a B-spline mesh, that is the mesh has a regular structure, the construction should exactly reproduce a B-spline surface. This goal is important for several reasons:

1. The algorithm is an extension of a widely used geometric modeling paradigm; its incorporation into geometric modeling tools might achieve broader acceptance than an entirely new method.
2. The algorithm can be used to deal with the irregularities of a nearly regular B-spline control mesh.
3. The patches generated over regular regions of a control mesh will be lower degree than those found by the general algorithm.

These reasons stem from both practical and theoretical concerns. In many design applications, control meshes may contain large regular regions, punctuated by irregularities to handle corners, branches, or to close off a solid object. By generalizing B-splines,

patches constructed over the regular regions of a control mesh will be lower degree than in the general case. This will result in a significant saving of time and space when dealing with many common control mesh topologies.

It is no coincidence that the geometric constructions presented here generate B-spline surfaces over regular control meshes. The ideas underlying many of the constructions given in Chapter 5 came from studying and generalizing the B-spline constructions of Chapter 3. As an interesting side effect, the generalized B-spline scheme can often generate lower depth S-patches even over semi-regular control meshes. In this chapter, it will be demonstrated that for certain control mesh restrictions, either on the order of the faces or the order of the vertices or both, reductions in the depth of the S-patches generated by the generalized scheme often occur. In the cases of regular rectangular or triangular control meshes, not only does the depth of the patches drop, it will be shown that the scheme is capable of generating tensor product bicubic and quartic triangular B-surfaces.

6.1 Degree Reductions

The degree reductions are based on the following formula given in Chapter 5

$$\overbrace{\text{depth}(H)}^7 \geq \max(\overbrace{\text{degree}(\mu_i)}^1 + \overbrace{\text{degree}(G_i)}^5, \overbrace{\text{degree}(\nu_i)}^3 + \overbrace{\text{degree}(\mathbf{D}_v G_i)}^3 + 1),$$

where the polynomial degree (or S-patch depth) of the various functions are written above them. If the value of one or more of $\text{degree}(\mu_i)$, $\text{degree}(G_i)$, $\text{degree}(\nu_i)$, or $\text{degree}(\mathbf{D}_v G_i)$ should happen to be less than the value shown, then the $\text{depth}(H)$ may actual be less than the general case value of 7. Such degree reductions take place as a result of local restrictions on the form of the control mesh. These restrictions and the resulting degree reductions are outlined in the following propositions.

The first proposition states that if the vertex orders of four consecutive vertices of a single face are equal, then the degree of the function ν_i associated with an edge of the face is in fact lower than the general value of 3.

PROPOSITION 6.1.1: Let $f = (\mathbf{v}_1, \dots, \mathbf{v}_n)$ be a face in \mathbf{F} . If $|\mathbf{v}_{i-1}| = |\mathbf{v}_i| = |\mathbf{v}_{i+1}| = |\mathbf{v}_{i+2}| = k$, then ν_i is quadratic.

PROOF : Replacing k_{i-1}, k_i, k_{i+1} , and k_{i+2} by k in Equations (5.50) through (5.53) and simplifying, one gets

$$\begin{aligned}\nu_{i,0} &= \sin \frac{2\pi}{k}, \\ \nu_{i,1} &= \frac{1}{3} \sin \frac{2\pi}{k} + \frac{2}{3} \tan \frac{\pi}{k} (1 - \cos \frac{2\pi}{n}), \\ \nu_{i,2} &= \frac{1}{3} \sin \frac{2\pi}{k} + \frac{2}{3} \tan \frac{\pi}{k} (1 - \cos \frac{2\pi}{n}), \\ \nu_{i,3} &= \sin \frac{2\pi}{k}.\end{aligned}$$

Given these values, it follows that

$$\begin{aligned}\nu_i(u) &= \nu_{i,0}B_0^3(u) + \nu_{i,1}B_1^3(u) + \nu_{i,2}B_2^3(u) + \nu_{i,3}B_3^3(u), \\ &= \nu_{i,0}(1-u)^3 + 3\nu_{i,1}(1-u)^2u + 3\nu_{i,2}(1-u)u^2 + \nu_{i,3}u^3, \\ &= \sin \frac{2\pi}{k} [(1-u)^3 + (1-u)^2u + (1-u)u^2 + u^3] \\ &\quad + 2 \tan \frac{\pi}{k} (1 - \cos \frac{2\pi}{n}) [(1-u)^2u + (1-u)u^2], \\ &= \sin \frac{2\pi}{k} [(1-u)^2 + u^2] + 2 \tan \frac{\pi}{k} (1 - \cos \frac{2\pi}{n}) [(1-u)u], \\ &= \sin \frac{2\pi}{k} B_0^2(u) + \tan \frac{\pi}{k} (1 - \cos \frac{2\pi}{n}) B_1^2(u) + \sin \frac{2\pi}{k} B_2^2(u).\end{aligned}$$

□

The control mesh restrictions of Proposition 6.1.1 indicate that whenever a face is made up of vertices of equal order, then the depth of the S-patch corresponding to the face will be less than 7. Precisely this type of restriction is present in the control meshes used by [Loop & DeRose 90]. In that scheme, faces were allowed to have arbitrary order while the order of vertices was required to be 4. Given these restriction, the resulting S-patches are at most depth 6.

Further reductions in the polynomial degree of the functions ν_i occur if, in addition to restricting local vertex orders, the order of a face is restricted. The second proposition demonstrates this fact.

PROPOSITION 6.1.2: If $|\mathbf{v}_{i-1}| = |\mathbf{v}_i| = |\mathbf{v}_{i+1}| = |\mathbf{v}_{i+2}| = k$ and $\cos \frac{2\pi}{k} = -\cos \frac{2\pi}{n}$, then ν_i is constant.

PROOF : By Proposition 6.1.1 it follows that

$$\begin{aligned}
\nu_i(u) &= \sin \frac{2\pi}{k} B_0^2(u) + \tan \frac{\pi}{k} (1 - \cos \frac{2\pi}{n}) B_1^2(u) + \sin \frac{2\pi}{k} B_2^2(u), \\
&= \sin \frac{2\pi}{k} B_0^2(u) + \frac{\sin \frac{2\pi}{k}}{1 + \cos \frac{2\pi}{k}} (1 + \cos \frac{2\pi}{k}) B_1^2(u) + \sin \frac{2\pi}{k} B_2^2(u), \\
&= \sin \frac{2\pi}{k} B_0^2(u) + \sin \frac{2\pi}{k} B_1^2(u) + \sin \frac{2\pi}{k} B_2^2(u), \\
&= \sin \frac{2\pi}{k}.
\end{aligned}$$

□

The control mesh restrictions of Proposition 6.1.2 are consistent with regular control meshes. A similar result holds for the functions μ_i as demonstrated in the third proposition.

PROPOSITION 6.1.3: If $|\mathbf{v}_i| = |\mathbf{v}_{i+1}| = k$ and $\cos \frac{2\pi}{k} = -\cos \frac{2\pi}{n}$, then μ_i is constant.

PROOF :

$$\begin{aligned}
\mu_i(u) &= \cos \frac{2\pi}{k} B_0^1(u) + [-\cos \frac{2\pi}{k} - 2\cos \frac{2\pi}{n}] B_1^1(u), \\
&= \cos \frac{2\pi}{k} B_0^1(u) + [-\cos \frac{2\pi}{k} + 2\cos \frac{2\pi}{k}] B_1^1(u), \\
&= \cos \frac{2\pi}{k} B_0^1(u) + \cos \frac{2\pi}{k} B_1^1(u), \\
&= \cos \frac{2\pi}{k}.
\end{aligned}$$

□

Propositions 6.1.2 and 6.1.3 indicate that for regular control meshes the functions μ_i and ν_i are actually constants. These results are used in the next section to help demonstrate the close relationship between the generalized B-spline scheme of this thesis and traditional regular B-spline schemes.

6.2 Generalizing B-spline Surfaces

In this section it is demonstrated that the general construction can be used to generate regular B-spline surfaces - in particular, bicubic tensor product B-splines and quartic

triangular B-splines. In both cases, it is shown that the boundary curves and transversal vector fields found by the constructions for these regular B-spline schemes are equivalent to those found by the general construction of this thesis for specific keypoints and shape parameters, and an appropriately restricted control mesh. It is also shown that the method for constructing the unknown interior points of S-patches is consistent with the regular B-spline constructions.

6.2.1 Bicubic B-spline Surfaces

It is now demonstrated that the general spline construction is equivalent to bicubic tensor product B-splines when, for all faces $f \in \mathbf{F}$ and vertices $\mathbf{v} \in \mathbf{V}$, the control mesh is restricted so that $|f| = |\mathbf{v}| = 4$. The equivalence of bicubic B-splines and the general construction also depends on setting the keypoints and shape parameter properly.

Let G be an edge map (as in Section 5.4) corresponding to a canonical edge in the restricted control mesh. Recall that $G(u)$ and $\mathbf{D}_v G(u)$, $u \in [0, 1]$ are a boundary curve and transversal vector field respectively. Let $Q(u)$ and $\mathbf{D}_v Q(u)$ represent the boundary curve and transversal vector field of a patch from a bicubic tensor product B-spline surface corresponding to same canonical edge as G . Let H be a 4-sided S-patch found by the construction of Chapter 5 such that some edge of $H(E(u))$ is constructed from the edge map G (here $E(u)$ maps $[0, 1]$ to an edge of H 's domain polygon). Recall that this means

$$H(E(u)) = G(u), \quad (6.1)$$

and

$$\mathbf{D}_v H(E(u)) = \mu(u)\mathbf{D}_u G(u) + \nu(u)\mathbf{D}_v G(E(u)), \quad (6.2)$$

where μ and ν are functions for this edge found in Section 5.5.1. Due to the mesh restrictions present in this case, Propositions 6.1.2 and 6.1.3 show that

$$\mu(u) = 0, \quad \text{and} \quad \nu(u) = 1.$$

Therefore the relationship between $\mathbf{D}_{\dot{v}} H(E(u))$ and $\mathbf{D} G(u)$ of Equation (6.2) simplifies to

$$\mathbf{D}_{\dot{v}} H(E(u)) = \mathbf{D}_{\dot{v}} G(u).$$

This indicates that $H(E(u)) = Q(u)$ and $\mathbf{D}_{\dot{v}} H(E(u)) = \mathbf{D}_{\dot{v}} Q(u)$ if

$$G(u) = Q(u), \tag{6.3}$$

and

$$\mathbf{D}_{\dot{v}} G(u) = \mathbf{D}_{\dot{v}} Q(u). \tag{6.4}$$

That is, the boundary curve and transversal vector field of an S-patch found by the general construction will be equivalent to those of a tensor product patch of a bicubic B-spline scheme if Equalities (6.3) and (6.4) hold. Once these equalities are established the patches found by the two schemes can only differ away from their boundaries. It will then be argued that this cannot be the case since the interior control point construction of Section 5.5.3 preserves bicubic behavior in this case.

Establishing Equalities (6.3) and (6.4) is relatively straightforward; the Bézier control points of $G(u)$ and $Q(u)$ and the control vectors of $\mathbf{D}_{\dot{v}} G(u)$ and $\mathbf{D}_{\dot{v}} Q(u)$ are determined in terms of control mesh vertices. Equality of corresponding control points (vectors) is equivalent to equality of curves (similarly transversal vector fields). Let the faces surrounding a single control mesh vertex be labeled as in Figure 6.1. The formulae for the Bézier control points of a bicubic B-spline patch from Section 3.2.1 (Equations (3.4) through (3.6)) are rewritten in terms of this labeling as

$$\mathbf{b}_{00} = \frac{4}{9}\mathbf{v} + \frac{1}{9}[\mathbf{v}_1 + \mathbf{v}_2 + \mathbf{v}_3 + \mathbf{v}_4] + \frac{1}{36}[\mathbf{w}_1 + \mathbf{w}_2 + \mathbf{w}_3 + \mathbf{w}_4], \tag{6.5}$$

$$\mathbf{b}_{10} = \frac{4}{9}\mathbf{v} + \frac{2}{9}\mathbf{v}_1 + \frac{1}{9}[\mathbf{v}_2 + \mathbf{v}_4] + \frac{1}{18}[\mathbf{w}_1 + \mathbf{w}_4], \tag{6.6}$$

$$\mathbf{b}_{11} = \frac{4}{9}\mathbf{v} + \frac{2}{9}[\mathbf{v}_1 + \mathbf{v}_2] + \frac{1}{9}\mathbf{w}_1. \tag{6.7}$$

The points \mathbf{b}_{00} , \mathbf{b}_{10} , and \mathbf{b}_{11} are three of the Bézier control points of the tensor product bicubic patch corresponding to face $\{\mathbf{v}, \mathbf{v}^1, \mathbf{w}^1, \mathbf{v}^2\}$. The remaining control points are found similarly by symmetric formulae.

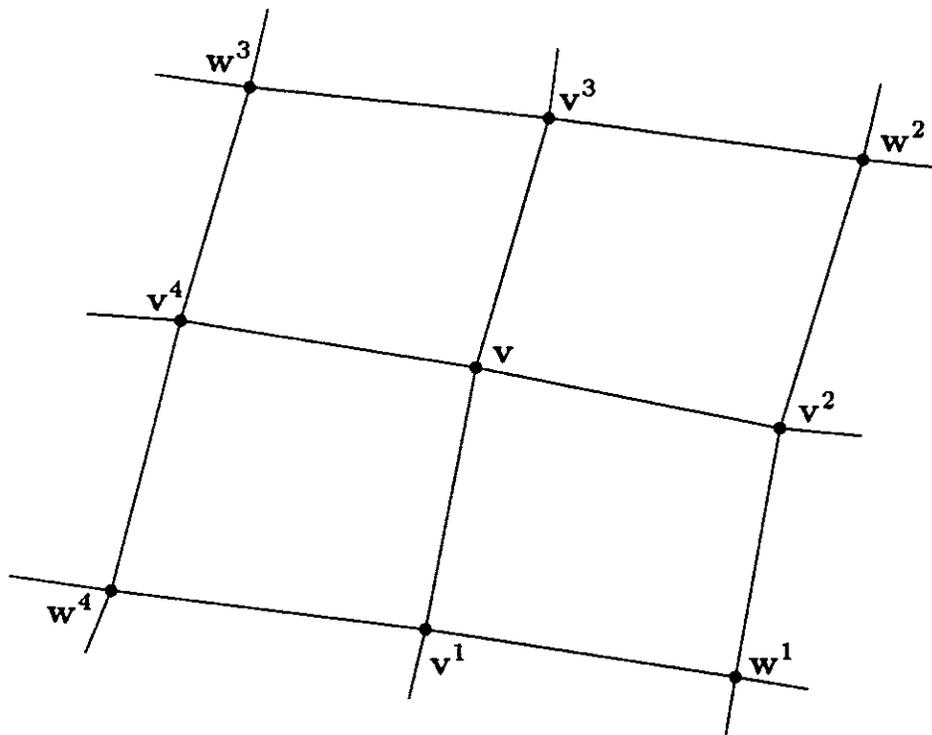


Figure 6.1: The labeling of a tensor product B-spline control mesh

A boundary curve of this patch is written as

$$Q(u) = \mathbf{b}_{00}B_0^3(u) + \mathbf{b}_{10}B_1^3(u) + \mathbf{b}_{20}B_2^3(u) + \mathbf{b}_{30}B_3^3(u).$$

A transversal vector field is written as

$$\mathbf{D}_v Q(u) = \dot{\mathbf{q}}_0 B_0^3(u) + \dot{\mathbf{q}}_1 B_1^3(u) + \dot{\mathbf{q}}_2 B_2^3(u) + \dot{\mathbf{q}}_3 B_3^3(u),$$

where

$$\dot{\mathbf{q}}_0 = 3(\mathbf{b}_{01} - \mathbf{b}_{00}),$$

$$\dot{\mathbf{q}}_1 = 3(\mathbf{b}_{11} - \mathbf{b}_{10}),$$

$$\dot{\mathbf{q}}_2 = 3(\mathbf{b}_{21} - \mathbf{b}_{20}),$$

$$\dot{\mathbf{q}}_3 = 3(\mathbf{b}_{31} - \mathbf{b}_{30}).$$

By substituting in Equations (6.5) through (6.7) and simplifying, these vectors are rewritten

$$\dot{\mathbf{q}}_0 = \frac{1}{3}[\mathbf{v}^2 - \mathbf{v}^4] + \frac{1}{12}[\mathbf{w}_1 + \mathbf{w}_2 - \mathbf{w}_3 - \mathbf{w}_4], \quad (6.8)$$

$$\dot{\mathbf{q}}_1 = \frac{1}{3}[\mathbf{v}^2 - \mathbf{v}^4] + \frac{1}{6}[\mathbf{w}_1 - \mathbf{w}_4]. \quad (6.9)$$

Similar formulae for $\dot{\mathbf{q}}_2$ and $\dot{\mathbf{q}}_3$ are found symmetrically.

Now consider the general construction. Recall that in the case of an edge $(\mathbf{v}, \mathbf{v}^1)$ where $|\mathbf{v}| = |\mathbf{v}^1| = 4$ the boundary curve $G(u)$ had 2 vector degrees of freedom not present in general. These degrees of freedom were set so that the boundary curve $G(u)$ would be cubic, rather than quintic. In this case, such a boundary curve can be written

$$G(u) = \mathbf{b}_{300}B_0^3(u) + \mathbf{b}_{210}B_1^3(u) + \mathbf{b}_{201}^1B_2^3(u) + \mathbf{b}_{300}^1B_3^3(u),$$

where \mathbf{b}_{300} and \mathbf{b}_{210} are Bézier control points from the vertex map (see Section 5.3) associated with vertex \mathbf{v} and \mathbf{b}_{300}^1 and \mathbf{b}_{201}^1 are Bézier control points from the vertex map associated with vertex \mathbf{v}^1 .

In the construction of the vertex map associated with vertex \mathbf{v} , let shape parameters $\alpha = 0$ and $\beta = 1$ (the default values for this case, see Section 5.3), and let the keypoint associated with vertex \mathbf{v} and face $\{\mathbf{v}, \mathbf{v}^1, \mathbf{w}^1, \mathbf{v}^2\}$ be found by

$$\mathbf{b}_{111} = \frac{4}{9}\mathbf{v} + \frac{2}{9}[\mathbf{v}_1 + \mathbf{v}_2] + \frac{1}{8}\mathbf{w}_1 + \frac{1}{72}[\mathbf{w}_3 - \mathbf{w}_2 - \mathbf{w}_4]. \quad (6.10)$$

This keypoint construction differs from the one given in the general case (see Equation (5.15)) and does not correspond to the “twist point” \mathbf{b}_{11} of a bicubic B-spline patch as one might expect. The reason for this apparent discrepancy has to do with the difference between the tensor product and S-patch representations of 4-sided patches (see Section 4.2).

Equation (5.16) from Section 5.4 for \mathbf{b}_{300} can be expanded for this case using (6.10) as follows:

$$\mathbf{b}_{300} = \frac{1}{4}\mathbf{b}_{111}^1 + \frac{1}{4}\mathbf{b}_{111}^2 + \frac{1}{4}\mathbf{b}_{111}^3 + \frac{1}{4}\mathbf{b}_{111}^4,$$

$$\begin{aligned}
&= \frac{1}{4}[\frac{4}{9}\mathbf{v} + \frac{2}{9}(\mathbf{v}_1 + \mathbf{v}_2) + \frac{1}{9}\mathbf{w}_1] + \cdots + \frac{1}{4}[\frac{4}{9}\mathbf{v} + \frac{2}{9}(\mathbf{v}_4 + \mathbf{v}_1) + \frac{1}{9}\mathbf{w}_4], \\
&= \frac{4}{9}\mathbf{v} + \frac{1}{9}[\mathbf{v}_1 + \mathbf{v}_2 + \mathbf{v}_3 + \mathbf{v}_4] + \frac{1}{36}[\mathbf{w}_1 + \mathbf{w}_2 + \mathbf{w}_3 + \mathbf{w}_4]. \tag{6.11}
\end{aligned}$$

This construction is identical to Equation (6.5).

Equation (5.17) from Section 5.4 is similarly expanded to get

$$\begin{aligned}
\mathbf{b}_{210} &= (0)\mathbf{v} + m_0\mathbf{b}_{111}^1 + m_1\mathbf{b}_{111}^2 + m_2\mathbf{b}_{111}^3 + m_3\mathbf{b}_{111}^4, \\
&= \frac{1}{2}\mathbf{b}_{111}^1 + \frac{1}{2}\mathbf{b}_{111}^4, \\
&= \frac{1}{2}[\frac{4}{9}\mathbf{v} + \frac{2}{9}(\mathbf{v}_1 + \mathbf{v}_2) + \frac{1}{8}\mathbf{w}_1 + \frac{1}{72}(\mathbf{w}_3 - \mathbf{w}_2 - \mathbf{w}_4)] \\
&\quad + \frac{1}{2}[\frac{4}{9}\mathbf{v} + \frac{2}{9}(\mathbf{v}_4 + \mathbf{v}_1) + \frac{1}{8}\mathbf{w}_4 + \frac{1}{72}(\mathbf{w}_2 - \mathbf{w}_1 - \mathbf{w}_3)], \\
&= \frac{4}{9}\mathbf{v} + \frac{2}{9}\mathbf{v}_1 + \frac{1}{9}[\mathbf{v}_2 + \mathbf{v}_4] + \frac{1}{18}[\mathbf{w}_1 + \mathbf{w}_4] \tag{6.12}
\end{aligned}$$

This construction is equivalent to Equation (6.6).

Equations (6.11) and (6.12) show that the Bézier control points \mathbf{b}_{00} and \mathbf{b}_{10} of $Q(u)$ are identical to \mathbf{b}_{300} and \mathbf{b}_{210} of $G(u)$ respectively. Similar symmetric results hold to show that $\mathbf{b}_{20} = \mathbf{b}_{201}^1$ and $\mathbf{b}_{30} = \mathbf{b}_{300}^1$. From this it can be concluded that $G(u) = Q(u)$. That is, the boundary curves constructed by the general scheme are identical to the boundary curves of tensor product bicubic B-splines (with appropriate shape parameters and keypoints).

The transversal vector field $\mathbf{D}_{\dot{i}} G(u)$ is

$$\mathbf{D}_{\dot{i}} G(u) = \dot{\mathbf{c}}_0 B_0^3(u) + \dot{\mathbf{c}}_1 B_1^3(u) + \dot{\mathbf{c}}_2 B_2^3(u) + \dot{\mathbf{c}}_3 B_3^3(u),$$

where

$$\dot{\mathbf{c}}_0 = 3(\mathbf{b}_{210} - \mathbf{b}_{300}), \tag{6.13}$$

$$\dot{\mathbf{c}}_1 = (\mathbf{b}_{201} + 2\mathbf{b}_{111}) - (\mathbf{b}_{300} + 2\mathbf{b}_{210}), \tag{6.14}$$

$$\dot{\mathbf{c}}_2 = (\mathbf{b}_{210}^1 + 2\mathbf{b}_{111}^1) - (\mathbf{b}_{300}^1 + 2\mathbf{b}_{201}^1), \tag{6.15}$$

$$\dot{\mathbf{c}}_3 = 3(\mathbf{b}_{210}^1 - \mathbf{b}_{300}^1), \tag{6.16}$$

are Equations (5.24) through (5.27) specialized to this regular case.

Equation (6.13) can be expanded to show that

$$\begin{aligned}
\dot{\mathbf{c}}_0 &= 3(\mathbf{b}_{210} - \mathbf{b}_{300}), \\
&= -\frac{4}{3}\mathbf{v} - \frac{1}{3}[\mathbf{v}_1 + \mathbf{v}_2 + \mathbf{v}_3 + \mathbf{v}_4] - \frac{1}{12}[\mathbf{w}_1 + \mathbf{w}_2 + \mathbf{w}_3 + \mathbf{w}_4] \\
&\quad + \frac{4}{3}\mathbf{v} - \frac{2}{3}\mathbf{v}^2 + \frac{1}{3}[\mathbf{v}_1 + \mathbf{v}_3] - \frac{1}{6}[\mathbf{w}_1 + \mathbf{w}_2], \\
&= \frac{1}{3}[\mathbf{v}^2 - \mathbf{v}^4] + \frac{1}{12}[\mathbf{w}_1 + \mathbf{w}_2 - \mathbf{w}_3 - \mathbf{w}_4].
\end{aligned} \tag{6.17}$$

This construction is identical to Equation (6.8).

Similarly Equation (6.14) can be expanded to show that

$$\begin{aligned}
\dot{\mathbf{c}}_1 &= (\mathbf{b}_{201} + 2\mathbf{b}_{111}) - (\mathbf{b}_{300} + 2\mathbf{b}_{210}), \\
&= -\frac{4}{9}\mathbf{v} - \frac{1}{9}[\mathbf{v}_1 + \mathbf{v}_2 + \mathbf{v}_3 + \mathbf{v}_4] - \frac{1}{36}[\mathbf{w}_1 + \mathbf{w}_2 + \mathbf{w}_3 + \mathbf{w}_4] \\
&\quad - \frac{8}{9}\mathbf{v} - \frac{4}{9}\mathbf{v}_1 - \frac{2}{9}[\mathbf{v}_2 + \mathbf{v}_4] - \frac{1}{9}[\mathbf{w}_1 + \mathbf{w}_4] \\
&\quad + \frac{4}{9}\mathbf{v} + \frac{2}{9}\mathbf{v}_2 + \frac{1}{9}[\mathbf{v}_1 + \mathbf{v}_3] + \frac{1}{18}[\mathbf{w}_1 + \mathbf{w}_2] \\
&\quad + \frac{8}{9}\mathbf{v} + \frac{4}{9}[\mathbf{v}_1 + \mathbf{v}_2] + \frac{1}{4}\mathbf{w}_1 + \frac{1}{36}[\mathbf{w}_3 - \mathbf{w}_2 - \mathbf{w}_4], \\
&= \frac{1}{3}[\mathbf{v}^2 - \mathbf{v}^4] + \frac{1}{6}[\mathbf{w}_1 - \mathbf{w}_4].
\end{aligned} \tag{6.18}$$

This construction is identical to Equation (6.9).

Equations (6.17) and (6.18) show that the Bézier control vectors $\dot{\mathbf{q}}_0$ and $\dot{\mathbf{q}}_1$ of $\mathbf{D}_i Q(u)$ are identical to $\dot{\mathbf{c}}_0$ and $\dot{\mathbf{c}}_1$ of $\mathbf{D}_i G(u)$ respectively. Similar symmetric results hold to show that $\dot{\mathbf{q}}_2 = \dot{\mathbf{c}}_2$ and $\dot{\mathbf{q}}_3 = \dot{\mathbf{c}}_3$. From this it can be concluded that $\mathbf{D}_i G(u) = \mathbf{D}_i Q(u)$. That is, the transversal vector fields constructed by the general scheme (with appropriate shape parameters and keypoints) are identical to the transversal vector fields of tensor product bicubic B-splines.

All that remains to demonstrate the equivalence of the patches constructed by the two schemes is to show that the interior points found for the S-patch in Section 5.5.3 are consistent with a degree elevated bicubic patch. The following argument should suffice: in Step 1 of the interior point construction, the control net of a depth 4 auxiliary patch is inscribed on a face of the control mesh; in Steps 2 and 3, the boundary panels of this auxiliary patch are modified to approximate the boundary behavior of the real S-patch

H ; in this special regular case, this approximation is in fact an interpolation (i.e. the boundary behavior of the auxiliary patch and the real S-patch are identical); the only point of the auxiliary patch not effected by Steps 2 and 3 is a single point at the centroid of the face. This point is precisely where it needs to be for the auxiliary patch to be identical to the regular bicubic B-spline patch; therefore, after Step 4 the resulting patch is still identical to a bicubic B-spline patch.

6.2.2 Quartic Triangular B-spline Surfaces

In this section, it is demonstrated that the general spline construction is equivalent to quartic triangular B-splines when, for all control mesh faces $f \in \mathbf{F}$ and control mesh vertices $\mathbf{v} \in \mathbf{V}$, $|f| = 3$ and $|\mathbf{v}| = 6$. Again, the equivalence depends on keypoint placement and the values of available shape parameters.

Let G be an edge map corresponding to a canonical edge of the regular triangular control mesh. $G(u)$ and $\mathbf{D}_{\dot{v}}G(u)$ are the boundary curve and transversal vector field respectively. Let $Q(u)$ and $\mathbf{D}_{\dot{v}}Q(u)$ represent the boundary curve and transversal vector field of a patch from a quartic triangular B-spline surface corresponding to the same canonical edge as G . Let H be the 3-sided S-patch (Bézier triangle) found by the generalized B-spline construction, such that $H(E(u))$ is constructed from edge map G . This means that G and H are related by Equations (6.1) and (6.2). Just as before, the mesh restrictions present in this case are consistent with Propositions 6.1.2 and 6.1.3 and show that

$$\mu(u) = 0, \quad \text{and} \quad \nu(u) = 1.$$

Therefore the relationship between $\mathbf{D}_{\dot{v}}H(E(u))$ and $\mathbf{D}G(u)$ again simplifies to

$$\mathbf{D}_{\dot{v}}H(E(u)) = \mathbf{D}_{\dot{v}}G(u),$$

indicating that $H(E(u)) = Q(u)$ and $\mathbf{D}_{\dot{v}}H(E(u)) = \mathbf{D}_{\dot{v}}Q(u)$ if Equations (6.3) and (6.4) are again satisfied.

Let the faces surrounding a single control mesh vertex be labeled as in Figure 6.2. The formulae for the Bézier control points of a quartic triangular B-spline surface from

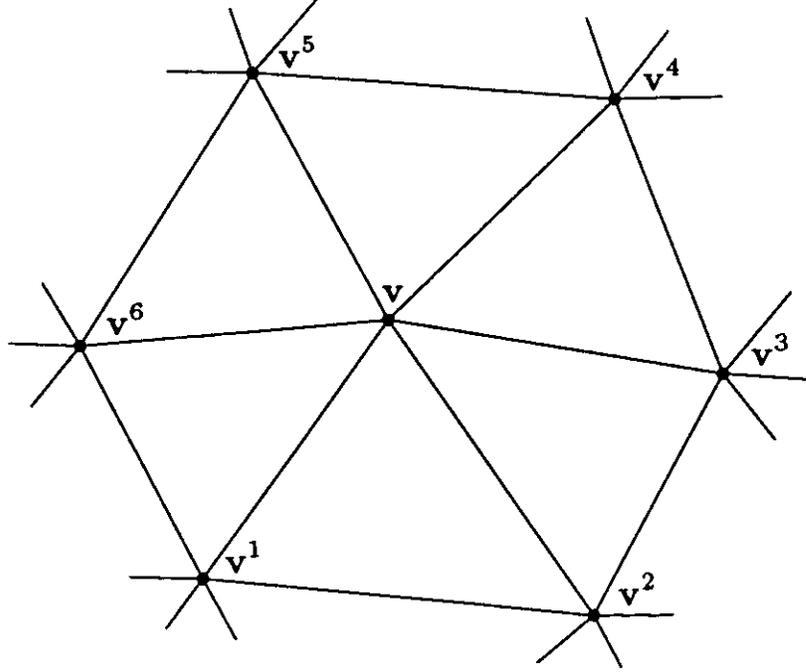


Figure 6.2: The labeling of a triangular B-spline control mesh

Section 3.3.1 (Equations (3.7) through (3.10)) are rewritten in terms of this labeling as

$$\mathbf{b}_{400} = \frac{1}{2}\mathbf{v} + \frac{1}{12}\mathbf{v}^1 + \frac{1}{12}\mathbf{v}^2 + \frac{1}{12}\mathbf{v}^3 + \frac{1}{12}\mathbf{v}^4 + \frac{1}{12}\mathbf{v}^5 + \frac{1}{12}\mathbf{v}^6, \quad (6.19)$$

$$\mathbf{b}_{310} = \frac{1}{2}\mathbf{v} + \frac{1}{6}\mathbf{v}^1 + \frac{1}{8}\mathbf{v}^2 + \frac{1}{24}\mathbf{v}^3 + \frac{1}{24}\mathbf{v}^5 + \frac{1}{8}\mathbf{v}^6, \quad (6.20)$$

$$\mathbf{b}_{220} = \frac{1}{3}\mathbf{v} + \frac{1}{3}\mathbf{v}^1 + \frac{1}{6}\mathbf{v}^2 + \frac{1}{6}\mathbf{v}^6, \quad (6.21)$$

$$\mathbf{b}_{211} = \frac{5}{12}\mathbf{v} + \frac{1}{4}\mathbf{v}^1 + \frac{1}{4}\mathbf{v}^2 + \frac{1}{24}\mathbf{v}^3 + \frac{1}{24}\mathbf{v}^6. \quad (6.22)$$

The points \mathbf{b}_{400} , \mathbf{b}_{310} , \mathbf{b}_{220} , and \mathbf{b}_{211} are four of the Bézier control points of a quartic triangular patch corresponding to face $\{\mathbf{v}, \mathbf{v}_1, \mathbf{v}_2\}$. The remaining control points are found by symmetric formulae.

The boundary curve of a patch is, in this case, the quartic written

$$Q(u) = \mathbf{b}_{400}B_0^4(u) + \mathbf{b}_{310}B_1^4(u) + \mathbf{b}_{220}B_2^4(u) + \mathbf{b}_{130}B_3^4(u) + \mathbf{b}_{040}B_4^4(u).$$

Since Q is the edge of a quartic patch any derivative function $\mathbf{D}_{\dot{w}} Q(u)$, where \dot{w} is an arbitrary direction, is at most cubic. Therefore, a transversal vector field can be written

$$\mathbf{D}_{\dot{v}} Q(u) = \dot{\mathbf{q}}_0 B_0^3(u) + \dot{\mathbf{q}}_1 B_1^3(u) + \dot{\mathbf{q}}_2 B_2^3(u) + \dot{\mathbf{q}}_3 B_3^3(u),$$

where

$$\dot{\mathbf{q}}_0 = -\frac{4}{\sqrt{3}} \mathbf{b}_{400} - \frac{4}{\sqrt{3}} \mathbf{b}_{310} + \frac{8}{\sqrt{3}} \mathbf{b}_{301},$$

$$\dot{\mathbf{q}}_1 = -\frac{4}{\sqrt{3}} \mathbf{b}_{310} - \frac{4}{\sqrt{3}} \mathbf{b}_{220} + \frac{8}{\sqrt{3}} \mathbf{b}_{211},$$

$$\dot{\mathbf{q}}_2 = -\frac{4}{\sqrt{3}} \mathbf{b}_{220} - \frac{4}{\sqrt{3}} \mathbf{b}_{130} + \frac{8}{\sqrt{3}} \mathbf{b}_{121},$$

$$\dot{\mathbf{q}}_3 = -\frac{4}{\sqrt{3}} \mathbf{b}_{130} - \frac{4}{\sqrt{3}} \mathbf{b}_{040} + \frac{8}{\sqrt{3}} \mathbf{b}_{031}.$$

Substituting in Equations (6.19) through (6.22) and simplifying shows

$$\dot{\mathbf{q}}_0 = \frac{1}{2\sqrt{3}} \mathbf{v}_2 + \frac{1}{2\sqrt{3}} \mathbf{v}_3 - \frac{1}{2\sqrt{3}} \mathbf{v}_5 - \frac{1}{2\sqrt{3}} \mathbf{v}_6, \quad (6.23)$$

$$\dot{\mathbf{q}}_1 = \frac{5}{6\sqrt{3}} \mathbf{v}_2 + \frac{1}{6\sqrt{3}} \mathbf{v}_3 - \frac{1}{6\sqrt{3}} \mathbf{v}_5 - \frac{5}{6\sqrt{3}} \mathbf{v}_6. \quad (6.24)$$

Similar formulae for $\dot{\mathbf{q}}_2$ and $\dot{\mathbf{q}}_3$ can be found symmetrically.

Now consider the general construction. In general $G(u)$ is a quintic boundary curve defined by

$$G(u) = \mathbf{c}_0 B_0^5(u) + \mathbf{c}_1 B_1^5(u) + \mathbf{c}_2 B_2^5(u) + \mathbf{c}_3 B_3^5(u) + \mathbf{c}_4 B_4^5(u) + \mathbf{c}_5 B_5^5(u),$$

where (repeating formulae from Section 5.4)

$$\mathbf{c}_0 = \mathbf{b}_{300}, \quad (6.25)$$

$$\mathbf{c}_1 = \frac{2}{5} \mathbf{b}_{300} + \frac{3}{5} \mathbf{b}_{210}, \quad (6.26)$$

$$\mathbf{c}_2 = \frac{1}{10} \mathbf{b}_{300} + \frac{6}{10} \mathbf{b}_{210} + \frac{3}{10} \mathbf{b}_{120}. \quad (6.27)$$

The points \mathbf{b}_{300} , \mathbf{b}_{210} , and \mathbf{b}_{120} are Bézier control points from the vertex map associated with vertex \mathbf{v} . Formulae for \mathbf{c}_3 , \mathbf{c}_4 , and \mathbf{c}_5 are defined similarly. For $G(u) = Q(u)$ it must be that $G(u)$ is actual a quartic curve and that $G(u)$'s quartic Bézier control points coincide with the control points of $Q(u)$.

If $G(u)$ is quartic then it should be possible to write

$$G(u) = \mathbf{f}_0 B_0^4(u) + \mathbf{f}_1 B_1^4(u) + \mathbf{f}_2 B_2^4(u) + \mathbf{f}_3 B_3^4(u) + \mathbf{f}_4 B_4^4(u),$$

where

$$\mathbf{f}_0 = \mathbf{c}_0, \quad (6.28)$$

$$\mathbf{f}_1 = \frac{5}{4}\mathbf{c}_1 - \frac{1}{4}\mathbf{c}_0, \quad (6.29)$$

$$\mathbf{f}_2 = \frac{5}{3}\mathbf{c}_2 - \frac{5}{6}\mathbf{c}_1 + \frac{1}{6}\mathbf{c}_0. \quad (6.30)$$

The points \mathbf{f}_3 and \mathbf{f}_4 can be written symmetrically in terms of \mathbf{c}_4 and \mathbf{c}_5 . It must also be possible to write \mathbf{f}_2 in terms of \mathbf{c}_3 , \mathbf{c}_4 and \mathbf{c}_5 ; both forms must be true (and equivalent) if $G(u)$ is quartic. By substituting Equations (6.25) through (6.27), Equations (6.28) through (6.30) can be rewritten

$$\mathbf{f}_0 = \mathbf{b}_{300}, \quad (6.31)$$

$$\mathbf{f}_1 = \frac{1}{4}\mathbf{b}_{300} + \frac{3}{4}\mathbf{b}_{210}, \quad (6.32)$$

$$\mathbf{f}_2 = \frac{1}{2}\mathbf{b}_{210} + \frac{1}{2}\mathbf{b}_{120}. \quad (6.33)$$

For a regular triangular control mesh the keypoints \mathbf{b}_{111} are constructed by setting

$$\mathbf{b}_{111}^i = \frac{1}{3}\mathbf{v} + \frac{1}{3}\mathbf{v}^i + \frac{1}{3}\mathbf{v}^{i+1}. \quad (6.34)$$

This coincides with Construction (5.15) of Section 5.3. Assuming the default shape parameter values $\alpha = \frac{1}{4}$, $\beta = 1$, Constructions (5.16) through (5.19) are specialized to the regular triangular case as follows:

$$\begin{aligned} \mathbf{b}_{300} &= \frac{1}{4}\mathbf{v} + \frac{1}{8}\mathbf{b}_{111}^1 + \frac{1}{8}\mathbf{b}_{111}^2 + \cdots + \frac{1}{8}\mathbf{b}_{111}^6, \\ &= \frac{1}{4}\mathbf{v} + \frac{1}{8}[\frac{1}{3}\mathbf{v} + \frac{1}{3}\mathbf{v}^1 + \frac{1}{3}\mathbf{v}^2] + \cdots + \frac{1}{8}[\frac{1}{3}\mathbf{v} + \frac{1}{3}\mathbf{v}^6 + \frac{1}{3}\mathbf{v}^1], \\ &= \frac{1}{2}\mathbf{v} + \frac{1}{12}\mathbf{v}^1 + \frac{1}{12}\mathbf{v}^2 + \frac{1}{12}\mathbf{v}^3 + \frac{1}{12}\mathbf{v}^4 + \frac{1}{12}\mathbf{v}^5 + \frac{1}{12}\mathbf{v}^6. \end{aligned} \quad (6.35)$$

$$\mathbf{b}_{210} = \frac{1}{4}\mathbf{v} + m_0\mathbf{b}_{111}^1 + m_1\mathbf{b}_{111}^2 + m_2\mathbf{b}_{111}^3 + m_3\mathbf{b}_{111}^4 + m_4\mathbf{b}_{111}^5 + m_5\mathbf{b}_{111}^6,$$

$$\begin{aligned}
&= \frac{1}{4}\mathbf{v} + \frac{7}{24}\mathbf{b}_{111}^1 + \frac{1}{8}\mathbf{b}_{111}^2 - \frac{1}{24}\mathbf{b}_{111}^3 - \frac{1}{24}\mathbf{b}_{111}^4 + \frac{1}{8}\mathbf{b}_{111}^5 + \frac{7}{24}\mathbf{b}_{111}^6, \\
&= \frac{1}{4}\mathbf{v} + \frac{7}{24}[\frac{1}{3}\mathbf{v} + \frac{1}{3}\mathbf{v}^1 + \frac{1}{3}\mathbf{v}^2] + \cdots + \frac{7}{24}[\frac{1}{3}\mathbf{v} + \frac{1}{3}\mathbf{v}^6 + \frac{1}{3}\mathbf{v}^1], \\
&= \frac{1}{2}\mathbf{v} + \frac{7}{36}\mathbf{v}^1 + \frac{5}{36}\mathbf{v}^2 + \frac{1}{36}\mathbf{v}^3 - \frac{1}{36}\mathbf{v}^4 + \frac{1}{36}\mathbf{v}^5 + \frac{7}{36}\mathbf{v}^6.
\end{aligned} \tag{6.36}$$

$$\begin{aligned}
\mathbf{b}_{120} &= (-1)\mathbf{b}_{210} + (1)\mathbf{b}_{111}^1 + (1)\mathbf{b}_{111}^6, \\
&= -[\frac{1}{2}\mathbf{v} + \frac{7}{36}\mathbf{v}^1 + \frac{5}{36}\mathbf{v}^2 + \cdots + \frac{7}{36}\mathbf{v}^6] + [\frac{1}{3}\mathbf{v} + \frac{1}{3}\mathbf{v}^1 + \frac{1}{3}\mathbf{v}^2] + [\frac{1}{3}\mathbf{v} + \frac{1}{3}\mathbf{v}^6 + \frac{1}{3}\mathbf{v}^1], \\
&= \frac{1}{6}\mathbf{v} + \frac{17}{36}\mathbf{v}^1 + \frac{7}{36}\mathbf{v}^2 - \frac{1}{36}\mathbf{v}^3 + \frac{1}{36}\mathbf{v}^4 - \frac{1}{36}\mathbf{v}^5 + \frac{5}{36}\mathbf{v}^6.
\end{aligned} \tag{6.37}$$

By substituting Equations (6.19) through (6.21), Equations (6.31) through (6.33) may be rewritten

$$\mathbf{f}_0 = \frac{1}{2}\mathbf{v} + \frac{1}{12}\mathbf{v}^1 + \frac{1}{12}\mathbf{v}^2 + \frac{1}{12}\mathbf{v}^3 + \frac{1}{12}\mathbf{v}^4 + \frac{1}{12}\mathbf{v}^5 + \frac{1}{12}\mathbf{v}^6, \tag{6.38}$$

$$\mathbf{f}_1 = \frac{1}{2}\mathbf{v} + \frac{1}{6}\mathbf{v}^1 + \frac{1}{8}\mathbf{v}^2 + \frac{1}{24}\mathbf{v}^3 + \frac{1}{24}\mathbf{v}^5 + \frac{1}{8}\mathbf{v}^6, \tag{6.39}$$

$$\mathbf{f}_2 = \frac{1}{3}\mathbf{v} + \frac{1}{3}\mathbf{v}^1 + \frac{1}{6}\mathbf{v}^2 + \frac{1}{6}\mathbf{v}^6. \tag{6.40}$$

A similar argument about the vertex \mathbf{v}^1 could be used to find \mathbf{f}_4 and \mathbf{f}_5 . This argument yields the same value for \mathbf{f}_2 , proving that $G(u)$ must be quartic. Since Equations (6.38) through (6.40) are identical to Equations (6.19) through (6.21), the quartic Bézier control points of $Q(u)$ and $G(u)$ are identical, hence $Q(u) = G(u)$. That is, the boundary curves of a quartic triangular B-spline patch and an S-patch from the general construction are identical for the specified keypoints and shape parameters.

The transversal vector field $\mathbf{D}_i G(u)$ is

$$\mathbf{D}_i G(u) = \dot{\mathbf{c}}_0 B_0^3(u) + \dot{\mathbf{c}}_1 B_1^3(u) + \dot{\mathbf{c}}_2 B_2^3(u) + \dot{\mathbf{c}}_3 B_3^3(u),$$

where

$$\dot{\mathbf{c}}_0 = -\frac{3}{\sqrt{3}}\mathbf{b}_{300} - \frac{3}{\sqrt{3}}\mathbf{b}_{210} + \frac{6}{\sqrt{3}}\mathbf{b}_{201}, \tag{6.41}$$

$$\dot{\mathbf{c}}_1 = -\frac{1}{\sqrt{3}}(\mathbf{b}_{300} + 2\mathbf{b}_{210}) - \frac{1}{\sqrt{3}}(\mathbf{b}_{210} + 2\mathbf{b}_{120}) + \frac{2}{\sqrt{3}}(\mathbf{b}_{201} + 2\mathbf{b}_{111}) \tag{6.42}$$

are Equations (5.24) and (5.25) specialized to the regular triangular case. Similar formulae exist for $\dot{\mathbf{c}}_2$ and $\dot{\mathbf{c}}_3$. Equations (6.41) and (6.42) may be expanded by substituting Equations (6.34) through (6.37). This will show that $\dot{\mathbf{c}}_0 = \dot{\mathbf{q}}_0$ and $\dot{\mathbf{c}}_1 = \dot{\mathbf{q}}_1$ (similarly that $\dot{\mathbf{c}}_2 = \dot{\mathbf{q}}_2$ and $\dot{\mathbf{c}}_3 = \dot{\mathbf{q}}_3$), thus $\mathbf{D}_v Q(u) = \mathbf{D}_v G(u)$. That is, the transversal vector field of a quartic triangular B-spline patch and an S-patch from the general construction are equivalent. Therefore, patches from these schemes can only differ away from patch boundaries.

A similar argument to the one used previously in the bicubic B-spline case shows that the interior points found for S-patches in Section 5.5.3 are consistent with a degree elevated quartic triangular B-spline patch. In Step 1 of the interior point construction all inscribed points belong to boundary panels; in Steps 2 and 3, the boundary panels of the auxiliary patch are set to approximate the behavior of the real S-patch H . Again in this special regular case, the approximation is an interpolation; therefore, after Steps 2 and 3 the auxiliary patch and real S-patch are identical. Step 4 only raises the degree of this patch, which is identical to a quartic triangular B-spline patch.

6.3 Other Special Cases

Propositions 6.1.1, 6.1.2, and 6.1.3 can be used to show that other special cases exist besides the bicubic tensor product and the quartic triangle B-spline specializations just presented. For example, whenever all of the vertices of a face are of the same order, the conditions of Proposition 6.1.1 are satisfied and the corresponding S-patch will drop from depth 7 to depth 6. Precisely this sort of restriction is imposed on the control mesh of a predecessor to the scheme presented here [Loop & DeRose 90]. In that scheme, all control vertices are required to be of order 4, although faces could be of arbitrary order. That scheme also generalized bicubic tensor product B-splines for regular control meshes, but in general generated depth 6 S-patches. The present scheme is also a generalization

Table 6.1: Reductions that occur for various mesh restrictions

$ \mathbf{v} $	$ f $	$\text{degree}(G)$	$\text{degree}(\mathbf{D}_v G)$	$\text{degree}(\mu)$	$\text{degree}(\nu)$	$\text{depth}(H)$
6	3	4	2	0	0	4
4	4	3	3	0	0	3
3	6	5	3	0	0	5
4	*	3	3	1	2	6
k	*	5	3	1	2	6
*	*	5	3	1	3	7

of that earlier scheme, when all control mesh vertices are of order 4¹.

The reductions in the depths of S-patches constructed by the generalized B-spline scheme for various control mesh restrictions are summarized in Table 6.3. Strictly speaking, the cases $|\mathbf{v}| = 6, |f| = 3$ and $|\mathbf{v}| = |f| = 4$ (the regular B-spline cases) would in general result in depth 5 patches. However, the lower depth results are shown since they have been shown to exist. Note that in the case of a regular hexagonal control mesh $|\mathbf{v}| = 3, |f| = 6$, the general construction generates depth 5 S-patches. It is likely that keypoints and shape parameters exist for this case that result in lower depth S-patches.

¹The interior point construction presented here is slightly different than that of [Loop & DeRose 90]

Chapter 7

Conclusions

The motivation for this work has been to find mathematical connections between various B-spline surfaces and to extend the topological range of surfaces that schemes like these can be used to design and represent. It is doubtful that B-spline theorists would call the surfaces created by the methods of this thesis true B-splines. True B-splines have many characteristics that the surfaces presented here do not share. Among these are subdivision, convolution, and a recursive definition that these generalized B-splines do not possess. It may well be that the great many intriguing and elegant properties of B-splines are artifacts of the regular control mesh with which they are associated. If so, it is doubtful whether true B-splines defined over irregular control meshes really exist. In the absence of such an elegant and general theory, the scheme proposed here serves as a practical attempt to fill this void.

Generalized B-spline surfaces based on S-patches have advantages over the subdivision surfaces of Catmull and Clark [Catmull & Clark 78] which are also a generalization of bicubic tensor product B-splines (although not a generalization of quartic triangular B-splines). The Catmull/Clark surfaces are defined algorithmically, not analytically. Evaluation and analysis of a surface and its geometric characteristics such as normals and curvatures is considerable more difficult for subdivision surfaces than for an analytically defined S-patch based surface. Furthermore, a patch based system allows for

a pipelined software design where each pipeline stage can work on individual surface elements without regard for the surface as a whole. Tessellation into polygons can be put off until absolutely needed, or avoided altogether.

In an interactive design environment it is highly computationally intensive to display in real-time the surface as one or more control mesh vertices are manipulated. It is quite feasible on the other hand to display just the boundary curves of patches. These additional shape queues can help give a designer a much better understanding of the surface being created than the control mesh alone. A system based on subdivision surfaces would require significantly more computational overhead to display the analogous boundary information in real-time.

Qualitatively, generalized B-splines based on S-patches generate smooth, well-behaved, free form surfaces. Quantitatively, however, there is a serious drawback encountered when dealing with S-patches. Compared to other parametric surface schemes based on triangular or tensor product patches, an S-patch based scheme is considerably less efficient in terms of storage space and evaluation time.

The number of control points needed to define an n -sided depth d S-patch is

$$\binom{n+d-1}{d}.$$

Specific examples for various values of n and d are given in Table 7. Evaluating an S-patch using deCasteljau's algorithm (Section 2.2.2) requires roughly

$$n \binom{n+d}{d}$$

vector additions and vector-scalar multiplications. Therefore, the complexity of deCasteljau's algorithm is $O(d^n)$. There are more efficient, albeit less stable, algorithms for evaluating S-patches (i.e., multivariate polynomials). At best however, a more efficient algorithm will lower this complexity by only one order of magnitude. Even if an upper limit is placed on depth (this limit is 7 here) the amount of time and space required for evaluating and storing S-patches is an exponential function of the number of sides. This is rather unfortunate and may prevent S-patches in general, and this generalized

Table 7.1: The number of S-patch control points for a given number of sides (down) and depth (across)

	1	2	3	4	5	6	7
3	3	6	10	15	21	28	36
4	4	10	20	35	56	84	120
5	5	15	35	70	126	210	330
6	6	21	56	126	252	462	792
7	7	28	84	210	462	924	1716
8	8	36	120	330	792	1716	3432

B-spline scheme in particular, from seeing wide spread use. These facts, perhaps more than any others, underscore the importance of the generalizing nature of this work. If patches with 5 or more sides are the exception, rather than the rule, then the efficiency concerns become somewhat less important.

Further research may be able to eliminate this dilemma. Recall that only the boundary panels play a role in determining a smooth join among S-patches. The interior control points, which in general constitute the majority of the control points, are needed because the Bernstein basis (Section 2.2) and the deCasteljau algorithm require them. A reasonable basis that is a subspace of the Bernstein basis could perhaps be found whose coefficients are exactly the S-patch boundary panels. The difficult part of finding such reasonable a basis is making sure that the corresponding surfaces have good shape. If such a basis does exist, only dn^2 control points would be needed to store each n -sided patch.

More significant than space considerations is evaluation time. If an algorithm of sufficiently low complexity for evaluating a surface based solely on boundary panels were found, say $O(dn^2)$, then the surface scheme proposed in this thesis would be extremely viable. In fact, such an algorithm can be formulated as follows: Associate a rational weight of zero with each interior point and a weight of one with each boundary panel point. Run deCasteljau's algorithm on the points with corresponding weights treated as an extra dimension, but do no work when a zero weight point is encountered. Finally,

normalize the output of deCasteljau’s algorithm, i.e. divide the point result by the weight result. By carefully avoiding steps that involve only zero weight points, this modified deCasteljau algorithm has the desired complexity. Unfortunately, the resulting surfaces tend to have hollow regions in the interiors of patches. More work is needed to determine if an algorithm like this can be found that generates aesthetically pleasing surfaces.

Additional work is also needed for determining keypoint placement (Section 5.3.2). In an interactive design system, these points would give users a great deal of control to fine tune a given shape. However, the default placement of keypoints should give good results in the majority of cases. A more careful study of how these points affect the shape of a surface could provide additional insights into keypoint placement algorithms. Alternatively, non-linear optimization techniques could be brought to bare to find keypoints that result in a best surface, for some reasonable definition of best.

The S-patches constructed here are in one-to-one correspondence with the faces of the control mesh. Alternatively, G^1 surfaces can be constructed from an arbitrary control mesh where the S-patches are in one-to-one correspondence with the *vertices* of the control mesh. Such a scheme would be a generalization of biquadratic B-spline surfaces. A partial solution to this *dual* problem is presented in [Loop & DeRose 90]. This solution required all control mesh faces f to have exactly 4 edges. The patches constructed in such dual schemes are generally of lower depth than patches of the corresponding *primal* schemes, i.e. schemes where patches correspond to faces. It may be that depth 6 S-patches suffice for a dual scheme without mesh restrictions. This question merits further investigation.

Bibliography

- [Barnhill et al. 73] R. Barnhill, G. Birkhoff, and W. Gordon. Smooth interpolation in triangles. *J. Approx. Theory*, 8(2):114–128, 1973.
- [Boehm 77] W. Boehm. Cubic B-spline curves and surfaces in computer aided geometric design. *Computing*, 19:29–34, 1977.
- [Boehm 80] W. Boehm. Inserting new knots into B-spline curves. *Computer Aided Design*, 12(4):199–201, 1980.
- [Boehm 81] W. Boehm. Generating the Bézier points of B-splines. *Computer Aided Design*, 13(6), 1981.
- [Boehm 82a] W. Boehm. The de Boor algorithm for triangular splines. In R. Barnhill and W. Boehm, editors, *Surfaces in Computer Aided Geometric Design*, pages 109–120. North-Holland, 1982.
- [Boehm 82b] W. Boehm. Generating the Bézier points of triangular splines. In R. Barnhill and W. Boehm, editors, *Surfaces in Computer Aided Geometric Design*, pages 77–91. North-Holland, 1982.
- [Boehm 83] W. Boehm. Subdividing multivariate splines. *Computer Aided Design*, 15(6):345–352, 1983.
- [Boehm 85a] W. Boehm. Calculating with box splines. *Computer Aided Geometric Design*, 18(2):149–162, 1985.

- [Boehm 85b] W. Boehm. Triangular spline algorithms. *Computer Aided Geometric Design*, 2(1-3):61–68, 1985.
- [Carpenter et al. 82] L. Carpenter, A. Fournier, and D. Fussell. Computer rendering of stochastic models. *Communications of the ACM*, 25(7):371–384, 1982.
- [Catmull & Clark 78] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer Aided Design*, 10(6):350–355, 1978.
- [Charrot & Gregory 84] P. Charrot and J. Gregory. A pentagonal surface patch for computer aided geometric design. *Computer Aided Geometric Design*, 1(1), 1984.
- [Chiyokura & Kimura 83] H. Chiyokura and F. Kimura. Design of solids with free-form surfaces. *Computer Graphics*, 17(3):289–298, 1983.
- [Dæhlen & Lyche 91] M. Dæhlen and T. Lyche. Box splines and approximations. In H. Hagen and D. Roller, editors, *Geometric Modeling: Methods and Applications*, pages 35–93. Springer-Verlag, Berlin, 1991.
- [Dahmen & Micchelli 84] W. Dahmen and C. Micchelli. Subdivision algorithms for the generation of box spline surfaces. *Computer Aided Geometric Design*, 18(2):115–129, 1984.
- [de Boor 87] C. de Boor. B-form basics. In G. Farin, editor, *Geometric Modeling: Algorithms and New Trends*, pages 131–148. SIAM, 1987.
- [de Casteljau 63] P. de Casteljau. Courbes et surfaces à poles. Technical report, A. Citroen, Paris, 1963.
- [de Casteljau 86] P. de Casteljau. *Shape Mathematics and CAD*. Kogan Page, London, 1986.

- [DeRose 85] A. DeRose. *Geometric Continuity: A Parametrization Independent Measure of Continuity for Computer Aided Geometric Design*. PhD dissertation, Berkeley, 1985. also tech report UCB/CSD 86/255.
- [DeRose 89] T. D. DeRose. A coordinate-free approach to geometric programming. In *Math for Siggraph*. Siggraph Course Notes #23, 1989. Also available as Technical Report No. 89-09-16, Department of Computer Science and Engineering, University of Washington, Seattle, WA (September, 1989).
- [Doo & Sabin 78] D. Doo and M. Sabin. Behaviour of recursive division surfaces near extraordinary points. *Computer Aided Design*, 10(6):356–360, 1978.
- [Farin 82] G. Farin. A construction for the visual C^1 continuity of polynomial surface patches. *Computer Graphics and Image Processing*, 20:272–282, 1982.
- [Farin 83] G. Farin. Algorithms for rational Bézier curves. *Computer Aided Design*, 15(2):73–77, 1983.
- [Farin 90] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design*. Academic Press, second edition, 1990.
- [Goodman 91] T. N. T. Goodman. Closed biquadratic surfaces. *Constructive Approximation*, 7(2):149–160, 1991.
- [Gregory 74] J. A. Gregory. Smooth interpolation without twist constraints. In R. E. Barnhill and R. F. Riesenfeld, editors, *Computer Aided Geometric Design*, pages 71–88. Academic Press, 1974.
- [Jensen 87] T. Jensen. Assembling triangular and rectangular patches and multivariate splines. In G. Farin, editor, *Geometric Modeling: Algorithms and New Trends*, pages 203–222. SIAM, 1987.

- [Lane & Riesenfeld 80] J. Lane and R. Riesenfeld. A theoretical development for the computer generation and display of piecewise polynomial surfaces. *IEEE Trans. Pattern Anal. Machine Intell.*, 2(1):35–46, 1980.
- [Lee & Majid 91] S. L. Lee and A. A. Majid. Closed smooth piecewise bicubic surfaces. *ACM Transactions on Graphics*, 10(4):342–365, 1991.
- [Loop & DeRose 89] C. T. Loop and T. D. DeRose. A multisided generalization of Bézier surfaces. *ACM Transactions on Graphics*, 8(3):204–234, July 1989.
- [Loop & DeRose 90] C. T. Loop and T. D. DeRose. Generalized B-spline surfaces of arbitrary topology. In *SIGGRAPH '90 Proceedings*, pages 347–356, 1990.
- [Mann et al. 92] S. Mann, C. Loop, M. Lounsbery, D. Meyers, J. Painter, T. DeRose, and K. Sloan. A survey of parametric scattered data fitting using triangular interpolants. In H. Hagen, editor, *Curve and Surface Modeling*. SIAM, 1992.
- [Nielson 79] G. Nielson. The side-vertex method for interpolation in triangles. *J of Approx Theory*, 25:318–336, 1979.
- [Peters 91] J. Peters. Smooth interpolation of a mesh of curves. *Constructive Approximation*, 7:221–246, 1991.
- [Piper 87] B. Piper. Visually smooth interpolation with triangular Bézier patches. In G. Farin, editor, *Geometric Modeling: Algorithms and New Trends*, pages 221–234. SIAM, 1987.
- [Ramshaw 87] L. Ramshaw. Blossoming: A connect-the-dots approach to splines. Technical report, Digital Systems Research Center, Palo Alto, Ca, 1987.
- [Ramshaw 88] L. Ramshaw. Béziers and b-splines as multiaffine maps. In R. Earnshaw, editor, *Theoretical Foundations of Computer Graphics and CAD*, pages 757–776. Springer Verlag, 1988.

- [Ramshaw 89] L. Ramshaw. Blossoms are polar forms. Technical report, Digital Systems Research Center, Palo Alto, Ca, 1989.
- [Sabin 76] M. Sabin. *The use of piecewise forms for the numerical representation of shape*. PhD dissertation, Hungarian Academy of Sciences, Budapest, Hungary, 1976.
- [Sablonnière 78] P. Sablonnière. Spline and Bézier polygons associated with a polynomial spline curve. *Computer Aided Design*, 10(4):257–261, 1978.
- [Shirman & Séquin 88] L. Shirman and C. Séquin. Local surface interpolation with Bézier patches. *Computer Aided Geometric Design*, 4(4):279–296, 1988.
- [van Wijk 86] J. van Wijk. Bicubic patches for approximating non-rectangular control-point meshes. *Computer Aided Geometric Design*, 3(1):1–13, 1986.
- [Watkins 88] M. A. Watkins. Problems in geometric continuity. *Computer Aided Design*, 20(8), 1988.