1. Motivation

Subdivision algorithms are similar to fractal procedures. In the standard fractal algorithm, we begin with a compact set $C_0$ and iterate over a collection of contractive transformations $W$ to generate a sequence of compact sets $C_{n+1} = W(C_n)$ that converge in the limit to a fractal shape $C_\infty = \lim_{n \to \infty} C_n$. In subdivision procedures, we start with a set $P_0$ (usually either a control polygon or a control polyhedron, or as we shall see shortly a quadrilateral or triangular mesh) and recursively apply a set of rules $S$ to generate a sequence of sets $P_{n+1} = S(P_n)$ of the same general type as $P_0$ that converge in the limit to a smooth curve or surface $P_\infty = \lim_{n \to \infty} P_n$.

Although one of the themes of Lecture 29 is that subdivision algorithms are essentially fractal procedures, nevertheless the goals of subdivision algorithms and fractal procedures are fundamentally different. The goal of fractal procedures is to construct extraordinary shapes, unconventional forms from conventional origins; the goal of subdivision algorithms is to construct smooth shapes, differentiable functions from discrete data. The de Casteljau subdivision algorithm for Bezier curves and surfaces and the Lane-Riesenfeld algorithm for uniform B-splines are examples of subdivision procedures that start with coarse control polygons or polyhedra and build refined control polygons and polyhedra that converge in the limit to smooth curves and surfaces.

We are interested in subdivision algorithms for four basic reasons:

i. Subdivision algorithms replace complicated formulas with simple procedures. Thus subdivision is more in the spirit of modern computer science than classical mathematics.

ii. Subdivision algorithms are easy to understand and simple to implement.

iii. Subdivision algorithms can generate a large class of smooth functions, not just polynomials and piecewise polynomials.

iv. Subdivision algorithms on polyhedral meshes can produce shapes with arbitrary topology, unlike tensor product schemes which can generate only surfaces that are topologically equivalent to a rectangle, or by identifying edges to a cylinder or a torus. Thus subdivision provides a simple approach to generating a wide variety of smooth shapes.

We are going to focus on subdivision algorithms for freeform surfaces. Most of this chapter is based on Chapters 2 and 7 of Subdivision Methods for Geometric Design: A Constructive Approach, by Warren and Weimer. We shall devote our attention here to the algorithmic details of different subdivision paradigms. Readers interested in proofs of convergence, differentiability, and other properties of these subdivision procedures should consult the book by Warren and Weimer.
2. Box Splines

Box spline surfaces are generalizations of uniform tensor product B-spline surfaces. The fundamental step in the Lane-Riesenfeld subdivision algorithm for uniform tensor product B-spline surfaces is averaging adjacent control points along canonical directions in a rectangular array. Box spline surfaces are generated by subdivision procedures that allow averaging along arbitrary directions in a rectangular array. To understand how this works in practice, we begin with a review of the Lane-Riesenfeld algorithm for uniform B-splines.

2.1 Split and Average. Recall that the Lane-Riesenfeld algorithm (Chapter 28, Section 4.1) for uniform B-spline curves consists of two basic steps: splitting and averaging (see Figure 1).

![Figure 1: The Lane-Riesenfeld algorithm for uniform B-spline curves. At the base of the diagram each control point is split into a pair of points. Adjacent points are then averaged and this averaging step is repeated \( n \) times, where \( n \) is the degree of the curve. The new control points for the spline with one new knot inserted midway between each consecutive pair of the original knots emerge at the top of the diagram. Iterating this procedure generates a sequence of control polygons that converge in the limit to the B-spline curve for the original control points.](image)

To extend the Lane-Riesenfeld knot insertion procedure to uniform tensor product B-spline surfaces \( P(s,t) \), we need to insert knots in both the \( s \) and \( t \) parameter directions. Consider a rectangular array of control points \( \{ P_{ij} \} \). To insert knots midway between each pair of consecutive knots, we first split and average in the \( s \)-direction, and then split and average the result of this computation in the \( t \)-direction (see Figures 2-4). Alternatively, instead of initially splitting each control point into two control points, we can start by splitting each control point into four control points. We can then average consecutively in the \( s \) and \( t \) directions; now there is no need to split the output of the averages in the \( s \)-direction before averaging in the \( t \)-direction (see Figure 5). The result of both approaches is the same control polyhedron, and it makes no difference whether we average first in the \( s \)-direction and then in the \( t \)-direction or first in the \( t \)-direction and then in the \( s \)-direction; the result is always the same (see Exercise 1a).
Figure 2: Subdivision for uniform tensor product B-spline surfaces. First step: (a) Split and (b) average in the s-direction. Here we illustrate the linear case. Higher degree surfaces can be generated by taking additional averages.

Figure 3: Subdivision for uniform tensor product B-spline surfaces continued. The output of the first step (Figure 2, right) is split -- that is, the points are doubled -- in the t-direction.
\[
\begin{align*}
P_0 & \quad P_0 + P_{12} & \quad \vdots & \quad P_0 + P_{12} \\
P_1 & \quad P_1 + P_{12} & \quad P_1 + P_{12} & \quad \vdots & \quad P_1 + P_{12} \\
\vdots & & \vdots & & \vdots & \vdots \\
P_0 & \quad P_0 + P_{12} & \quad \vdots & \quad P_0 + P_{12} \\
\end{align*}
\]

**Average in the \( t \)-direction**

**Figure 4:** Subdivision for uniform tensor product B-spline surfaces. Final step: Average the points in Figure 3 in the \( t \)-direction. Here we illustrate the bilinear case. Higher bidegree surfaces can be generated by taking additional averages.

\[
\begin{align*}
P_0 & \quad P_0 & \quad P_{12} & \quad P_{22} & \quad P_{22} & \quad \cdots \\
P_0 & \quad P_0 & \quad P_{12} & \quad P_{22} & \quad P_{22} & \quad \cdots \\
\vdots & & \vdots & & \vdots & \vdots \\
P_0 & \quad P_0 & \quad P_{12} & \quad P_{22} & \quad P_{22} & \quad \cdots \\
\end{align*}
\]

**Figure 5:** Subdivision for uniform tensor product B-spline surfaces -- alternative approach. First (\( a \)) split each point into four points and then (\( b \)) average in the \( s \)-direction. The resulting points are the same as the points in Figure 3. We can now average the output in the \( t \)-direction to get the points in Figure 4. No additional splitting is required.

4
2.2 A Subdivision Procedure for Box Spline Surfaces. Box spline surfaces are built by a generalization of the split and average approach to subdivision for uniform tensor product B-spline surfaces. The only difference is that for box splines, in addition to averaging along canonical directions in the rectangular array of control points, we also allow averaging along arbitrary directions in the rectangular array.

Thus to define a box spline surface, in addition to a rectangular array of control points \( \{P_{ij}\} \) in 3-space, we need to specify a collection of vectors \( V = \{v = (v_1, v_2)\} \) in the plane. The only restriction on the collection \( V \) is that the components \( v_1, v_2 \) of each of the vectors \( v \) must be integers, and \( V \) must contain at least one copy of the vectors \((1,0)\) and \((0,1)\). Note that vectors \( v \) in the collection \( V \) can appear multiple times.

The subdivision algorithm for a box spline surface begins by splitting each control point into four control points, just as in the subdivision algorithm for uniform tensor product B-spline surfaces (see Figure 5, left). We then remove one copy of the vectors \((1,0)\) and \((0,1)\) from the collection \( V \). Thus the initial copies of \((1,0)\) and \((0,1)\) correspond to quadrupling the control points. Now we make one averaging pass for each remaining vector \( v = (v_1, v_2) \) in the collection \( V \) by computing the averages of \( v \)-adjacent points -- that is, by computing all averages of the form:

\[
Q_{i,j} = \frac{P_{i,j} + P_{i+v_1,j+v_2}}{2}.
\]

The output of one averaging pass is used as input to the next averaging pass. The order in which we compute these averages does not matter, since averaging \( v \)-adjacent points and then averaging \( w \)-adjacent points is equivalent to averaging \( w \)-adjacent points and then averaging \( v \)-adjacent points (see Exercise 1b).

Iterating this procedure generates in the limit piecewise polynomial surfaces of degree \( d - 2 \), where \( d \) is the number of vectors, counting multiplicities, in the collection \( V \). Moreover these surfaces have \( \alpha - 2 \) continuous derivatives, where \( \alpha \) is the size of the smallest set \( A \subset V \) such that the vectors in \( V - A \) are all multiples of a single vector (see Warren and Weimer, Chapter 2).

**Example 2.1: Uniform Tensor Product B-Spline Surfaces**

Let \( V \) consist of the vector \((1,0)\) repeated \( m \) times and the vector \((0,1)\) repeated \( n \) times. The initial copies of \((1,0)\) and \((0,1)\) correspond to quadrupling the control points. Thus, repeating the vector \((1,0)\) \( m \) times corresponds to taking \( m - 1 \) averages in the \( s \)-direction, and repeating the vector \((0,1)\) \( n \) times corresponds to taking \( n - 1 \) averages in the \( t \)-direction. Therefore the corresponding box spline surface is a uniform tensor product B-spline surface of bidegree \((m-1,n-1)\). The total degree of this surface is \( d = (m-1) + (n-1) = m + n - 2 \) and this surface has \( \alpha - 2 \) continuous derivatives, where \( \alpha = \min(m,n) \).
Example 2.2: Three Direction Linear Box Splines

Let $V = \{(1,0),(0,1),(1,1)\}$. Then $d = 3$ and $\alpha = 2$. Thus the corresponding surface is a continuous, 3-direction, piecewise linear box spline. The vectors $(1,0)$ and $(0,1)$ correspond to quadrupling the control points. Averaging these points in the direction $(1,1)$ yields the points in Figure 6.

\[
\begin{array}{cccccc}
P_{02} & P_{02} + P_{12} & \frac{P_{12} + P_{22}}{2} & \cdots \\
\frac{P_{01} + P_{02}}{2} & \frac{P_{01} + P_{12}}{2} & \frac{P_{11} + P_{12}}{2} & \frac{P_{11} + P_{22}}{2} & \frac{P_{21} + P_{22}}{2} & \cdots \\
P_{01} & \frac{P_{01} + P_{11}}{2} & \frac{P_{11}}{2} & \frac{P_{11} + P_{21}}{2} & P_{21} & \cdots \\
\frac{P_{00} + P_{01}}{2} & \frac{P_{00} + P_{11}}{2} & \frac{P_{10} + P_{11}}{2} & \frac{P_{10} + P_{21}}{2} & \frac{P_{20} + P_{21}}{2} & \cdots \\
P_{00} & \frac{P_{00} + P_{10}}{2} & \frac{P_{10}}{2} & \frac{P_{10} + P_{20}}{2} & P_{20} & \cdots \\
\end{array}
\]

**Figure 6:** One level of subdivision for the 3-direction, piecewise linear box spline in Example 2.2. These control points are generated by averaging the control points in Figure 5, left, along the direction $(1,1)$ — that is, by using Equation (2.1) with $v_1 = v_2 = 1$. For an example, see Figure 8.

Example 2.3: Three Direction Quartic Box Splines

Let $V = \{(1,0),(1,1),(0,1),(0,0),(1,1),(1,1)\}$. Then $d = 6$ and $\alpha = 4$. Thus the corresponding surface is a 3-direction, piecewise quartic box spline with two continuous derivatives. The initial vectors $(1,0)$ and $(0,1)$ correspond to quadrupling the control points. Averaging these points once in each of the directions $(1,0)$ and $(0,1)$ yields the points in Figure 4. Averaging twice more in the direction $(1,1)$ generates the points in Figure 7 (see Exercise 2).

\[
\begin{array}{cccccc}
3P_{01} + P_{02} + P_{11} + 3P_{12} & 3P_{11} + 3P_{12} + P_{22} & 3P_{11} + P_{12} + P_{21} + 3P_{22} & \cdots \\
8 & 8 & 8 & \cdots \\
\frac{P_{00} + 3P_{01} + 3P_{11} + P_{12}}{8} & \frac{P_{00} + P_{01} + P_{10} + 10P_{11} + P_{12} + P_{21} + P_{22}}{16} & \frac{P_{10} + 3P_{11} + 3P_{21} + P_{22}}{8} & \cdots \\
3P_{00} + P_{01} + P_{10} + 3P_{11} & 3P_{10} + P_{10} + 3P_{11} + P_{21} & 3P_{10} + P_{11} + P_{20} + 3P_{21} & \cdots \\
8 & 8 & 8 & \cdots \\
\end{array}
\]

**Figure 7:** One level of subdivision for the 3-direction, quartic box spline in Example 2.3. These control points are generated by averaging the control points in Figure 4 twice along the direction $(1,1)$. For an example, see Figure 9.
**Figure 8:** A three direction piecewise linear box spline surface. The initial control polygon is on the left; the first two levels of subdivision are illustrated in the middle and on the right.

**Figure 9:** A three direction quartic box spline surface. The initial control polygon is in the upper left. The first five levels of subdivision are illustrated from left to right and from top to bottom.

**Figure 10:** The $C^1$ biquadratic B-spline (center) and the $C^2$ three direction quartic box spline (right) for the same control polygon (left). Both surfaces employ four rounds of averaging for each level of subdivision.
3. Quadrilateral Meshes

A quadrilateral mesh is a collection of quadrilaterals, where each pair of quadrilaterals are either disjoint or share a common edge and each edge belongs to at most two quadrilaterals. One way to form a quadrilateral mesh is to start from a rectangular array of control points. Joining points with adjacent indices generates a regular quadrilateral mesh, a mesh where each interior vertex has four adjacent faces, edges, and vertices (see Figures 8, 9, 10). Tensor product B-splines are built from rectangular arrays of control points. Therefore uniform tensor product B-splines are an example of surfaces generated by subdivision starting from a quadrilateral mesh.

The goal of this section is to construct smooth surfaces starting from quadrilateral meshes of arbitrary topology. Unlike tensor product or box spline schemes, the vertices of an arbitrary quadrilateral mesh are not constrained to form a rectangular array; all that is required is that the vertices can be joined into non-overlapping quadrilateral faces, where each pair of faces are either disjoint or share a common edge and each edge belongs to at most two faces. For example, the faces of a cube form a quadrilateral mesh, even though the vertices of the cube do not form a regular rectangular array, since in the cube each vertex has only three adjacent faces (see Figure 14). Two additional examples of quadrilateral meshes that are not generated by rectangular arrays of points are illustrated in Figure 13(a) and Figure 15.

Below we shall provide two methods for constructing smooth surfaces from arbitrary quadrilateral meshes via subdivision: centroid averaging and stencils.

3.1 Centroid Averaging. Centroid averaging is an extension of the Lane-Riesenfeld subdivision algorithm for uniform tensor product bicubic B-spline surfaces from rectangular arrays of control points to quadrilateral meshes with arbitrary topology. Therefore we begin by revisiting the Lane-Riesenfeld algorithm for bicubic B-splines.

3.1.1 Uniform Bicubic B-Spline Surfaces. To simplify our discussion, let us start with uniform B-spline curves. For cubic B-splines the Lane-Riesenfeld algorithm can be separated into two distinct stages: topology (connectivity) and geometry (shape). In the topological stage, corresponding to the first, piecewise linear step of the Lane-Riesenfeld algorithm, we introduce new control points at the midpoints of the original control polygon and we change the connectivity of the control polygon by adding edges joining these new points to adjacent control points -- see Figure 1 and Figure 11(b). Notice, however, that in the topological phase we do not alter the shape of the control polygon. In the geometric stage, we reposition the control points by taking two successive averages. Notice that computing two successive averages is equivalent to taking the midpoints of the midpoints of these control points -- see Figure 11(c). Thus in the geometric phase we change the shape of the control polygon, but we do not alter the connectivity of the vertices and edges.
Therefore each averages of four adjacent control points. Computing the centroid of adjacent centroids sends therefore two successive averages in the $s$-direction and the $t$-direction. But successive averaging in the $s$ and $t$ directions is equivalent to a single (bilinear) average of four adjacent control points, since

$$\frac{1}{4} \left( Q_{i-1,j-1} + Q_{i-1,j} + Q_{i-1,j+1} + Q_{i,j} \right) + \frac{1}{4} \left( Q_{i,j-1} + Q_{i,j} + Q_{i+1,j} + Q_{i+1,j+1} \right).$$

Therefore two successive averages in the $s$ and $t$ directions are equivalent to two successive centroid averages of four adjacent control points. Computing the centroid of adjacent centroids sends

$$Q_{ij} \rightarrow \frac{1}{4} \left( Q_{i-1,j-1} + Q_{i-1,j} + Q_{i-1,j+1} + Q_{i,j} \right) + \frac{1}{4} \left( Q_{i,j-1} + Q_{i,j} + Q_{i+1,j} + Q_{i+1,j+1} \right)$$

$$+ \frac{1}{4} \left( Q_{i-1,j} + Q_{i-1,j+1} + Q_{i,j} + Q_{i,j+1} \right) + \frac{1}{4} \left( Q_{i,j} + Q_{i+1,j} + Q_{i+1,j+1} + Q_{i+1,j+1} \right).$$

Therefore each control point is repositioned to the centroid of the centroids of the faces adjacent to

Figure 11: One level of the Lane-Riesenfeld algorithm for uniform cubic B-spline curves. In (a) a segment of the original control polygon (black) is illustrated. In the topological stage (b), new control points (yellow) are introduced at the midpoints of the edges of the control polygon. Notice that there are now two types of control points: edge points (yellow) and vertex points (black). In the geometric stage (c), the control points (black and yellow) are repositioned to the midpoints (blue) of the midpoints (green) of these control points. Edge points (yellow) are relocated along the same edge (in fact, to the same position along the edge), but vertex points (black) are relocated to new positions off the original control polygon. The new control polygon is illustrated in blue.

A similar interpretation applies to the Lane-Riesenfeld subdivision algorithm for uniform tensor product bicubic B-spline surfaces. Once again the Lane-Riesenfeld algorithm can be separated into two distinct stages: topology (connectivity) and geometry (shape).

In the topological stage, we keep the original control points and we insert new control points at the midpoints of the edges and at the centroids of the faces of the control polyhedron (see Figure 4 and Figure 12(b)). Thus we alter the connectivity of the control polyhedron, adding new edges and faces by connecting the centroid of each face to the midpoints of the edges surrounding the face. Notice again that in the topological phase, we do not alter the shape of the control polyhedron.

In the geometric stage, we reposition these control points by taking two successive averages in the $s$-direction and the $t$-direction. But successive averaging in the $s$ and $t$ directions is equivalent to a single (bilinear) average of four adjacent control points, since
the control point (see Figure 12(c)). Thus in the geometric stage we change the shape of the control polyhedron, but we do not alter the connectivity of the vertices, edges, and faces.

![Original Control Points](image1)
![New Topology](image2)
![New Geometry](image3)

**Figure 12:** One level of the Lane-Riesenfeld algorithm for bicubic B-spline surfaces. In (a) we illustrate one face of the control polyhedron. In the topological stage (b), new control points (yellow) are inserted at the midpoints of the edges and at the centroids of the faces of the control polyhedron; each edge is then split into two edges and each face is split into four faces. In the geometric stage (c), each control point -- yellow or black -- is repositioned to the centroid (blue) of the centroids (green) of the faces adjacent to the control point. Thus after the geometric stage, each black and each yellow control point is repositioned based on the location of the green centroids of adjacent faces, but the topology of the polyhedron is not changed. The green centroids are used only to reposition vertices and are not themselves vertices of the refined polyhedron.

### 3.1.2 Arbitrary Quadrilateral Meshes.

As with bicubic B-splines, there are two main phases to centroid averaging for arbitrary quadrilateral meshes: topology (connectivity) and geometry (shape). In the topological phase we refine the mesh by introducing new vertices, edges, and faces, but we do not alter the underlying shape of the mesh. In the geometric phase we reposition the vertices to change the geometry of the mesh, but we do not alter the underlying topology of the mesh; rather we maintain the connectivity of the mesh -- the edges and faces -- introduced during the topological phase.

To refine the topology of the mesh, we proceed exactly as we did for bicubic B-splines: we keep the original vertices and we insert new *edge vertices* at the midpoints of the edges and new *face vertices* at the centroids of the faces of the quadrilateral mesh. Each edge is then split into two edges and each face is split into four faces by connecting the centroid of the face to the midpoints of the edges surrounding the face (see Figure 13(b)).

To alter the geometry of the mesh, we reposition each vertex of the mesh to the centroid of the centroids of the adjacent faces. The only difference between subdivision for bicubic B-splines and subdivision for arbitrary quadrilateral meshes is that for arbitrary quadrilateral meshes there need not be four faces adjacent to each interior vertex. For example, in the cube there are only three faces adjacent to each of the vertices. An interior vertex of a quadrilateral mesh where the number of adjacent faces, edges, or vertices is not equal to four is called an *extraordinary vertex*, and the number of faces, edges, or vertices adjacent to a vertex is called the *valence* of the vertex. Thus in
the geometric phase, each vertex $Q$ of the mesh is repositioned by the formula
\[ Q \rightarrow \frac{1}{n} \sum_{k=1}^{n} C_k , \]
where $C_1, \ldots, C_n$ are the centroids of the faces adjacent to $Q$ and $n$ is the valence of $Q$.

After several levels of subdivision, most of the interior vertices of a quadrilateral mesh are ordinary vertices and the extraordinary vertices become more and more isolated topologically from one another. These features emerge because during the topological phase, each face is subdivided into four faces by edges joining the centroid of the face to the midpoints of the surrounding edges. Thus the new edge vertices and the new face vertices all have valence four (see Figure 13(b)), so all the new vertices introduced by subdivision are ordinary vertices. Also the valence of each of the original vertices is unchanged, since the number of faces surrounding a vertex does not change during subdivision. Hence during subdivision, ordinary vertices remain ordinary vertices and extraordinary vertices remain extraordinary vertices with the same valence (see Figure 13(b)). Since all the new vertices have valence four, locally the new quadrilateral mesh looks exactly like a rectangular array of control points; only vertices from the original mesh that did not have valence four do not have four adjacent faces. Therefore, just like uniform bicubic B-splines, the surfaces generated in the limit by iterating centroid averaging have two continuous derivatives everywhere; the only exceptions are at the limits of the extraordinary vertices, where these surfaces are guaranteed to have only one continuous derivative (see Warren and Weimer, Chapter 8).

![Figure 13: One level of centroid averaging. In (a) we illustrate six faces of a quadrilateral mesh surrounding an extraordinary vertex. In the topological stage (b), new vertices are inserted at the midpoints (yellow) of the edges and at the centroids (green) of the faces, and the centroid of each face is joined to the midpoints of the surrounding edges, splitting each face into four new faces. In the geometric stage (c), each vertex -- yellow, green, or black -- is repositioned to the centroid of the centroids (blue) of the faces adjacent to the vertex, but the topology of the mesh is not changed. Notice that the extraordinary vertex at the center remains an extraordinary vertex and is repositioned as the centroid of six adjacent centroids. All the other vertices -- black, yellow, and green -- are ordinary vertices and their new location is computed as the centroid of four blue points, just as in the Lane-Riesenfeld algorithm for bicubic B-splines.](image)
3.2 Stencils. Stencils are an alternative way to generate smooth surfaces via subdivision starting from quadrilateral meshes of arbitrary topology. Centroid averaging separates the computation of new vertices during subdivision into two phases: topology (connectivity) and geometry (shape). After the insertion of new vertices in the topological phase, all the vertices, old and new, are repositioned by a single formula -- centroid averaging. This formula computes the new positions of the vertices in terms of the known positions of the vertices after the topological phase. In contrast, stencils compute the positions of all the new vertices in terms of the positions of the original vertices. Thus stencils combine the topological and geometric computations into a single phase.
3.2.1 Stencils for Uniform B-Splines. Since centroid averaging is an extension of the Lane-Riesenfeld subdivision algorithm for tensor product bicubic B-spline surfaces from rectangular arrays of control points to quadrilateral meshes with arbitrary topology, we shall adapt stencils for bicubic B-spline surfaces to quadrilateral meshes of arbitrary topology. As usual, to simplify matters, we will begin our study of stencils with stencils for cubic B-spline curves.

The Lane-Riesenfeld algorithm for cubic B-spline curves is illustrated in Figure 16 (see too Chapter 28, Section 4.1).

Figure 16: The Lane-Riesenfeld algorithm for cubic B-spline curves. At the bottom of the diagram each control point is split into a pair of points. Adjacent points are then averaged and this averaging step is repeated three times. The new control points for the cubic spline with one new knot inserted midway between each consecutive pair of the original knots emerge at the top of the diagram.

Notice that there are essentially two distinct explicit formulas for the new control points that emerge at the top of this diagram:

\[ Q_{i+1/2} = \frac{P_i + P_{i+1}}{2} \]  \hspace{1cm} (3.1)

\[ Q_{i+1} = \frac{P_i + 6P_{i+1} + P_{i+2}}{8} \]  \hspace{1cm} (3.2)

The stencils for this algorithm are simply the coefficients \(1/2, 1/2\) and \(1/8, 6/8, 1/8\) that appear in these two formulas.

To understand the geometry behind these two formulas, recall from Figure 11 that after one
level of subdivision topologically the new control polygon for a cubic B-spline curve consists of two types of control points: edge points and vertex points. Equation (3.1) repositions edge points; Equation (3.2) repositions vertex points. These stencils can be illustrated schematically by the diagrams in Figure 17.

![Stencils](image)

**Figure 17:** Stencils for subdivision of cubic B-spline curves: (a) the edge stencil and (b) the vertex stencil. The position of the new control point (clear diamond) is computed by multiplying the original control points (black discs) by the associated fractions and summing the results.

There is a similar interpretation using stencils of the Lane-Riesenfeld subdivision algorithm for bicubic B-spline surfaces. Topologically there are now three kinds of points: face points, edge points, and vertex points. To find the stencils for these points, we need to find explicit formulas for the control points of the new control polyhedron after one level of subdivision. We can calculate these points by taking two rounds of centroid averaging of the control points in Figure 4. The results are presented in Figure 18.

\[
\begin{align*}
&\frac{9P_{01} + 3P_{02} + 3P_{11} + P_{12}}{16} & \frac{3P_{01} + P_{02} + 9P_{11} + 3P_{12}}{16} & \frac{9P_{11} + 3P_{12} + 3P_{21} + P_{22}}{16} \\
&\frac{3P_{00} + 9P_{01} + P_{10} + 3P_{11}}{16} & \frac{P_{00} + 3P_{01} + 3P_{10} + 9P_{11}}{16} & \frac{3P_{10} + 9P_{11} + P_{20} + 3P_{21}}{16} \\
&\frac{9P_{00} + 3P_{01} + 3P_{10} + P_{11}}{16} & \frac{3P_{00} + P_{01} + 9P_{10} + 3P_{11}}{16} & \frac{9P_{10} + 3P_{11} + 3P_{20} + P_{21}}{16}
\end{align*}
\]

(a) *First round of centroid averaging*
\[
\begin{align*}
\frac{P_{00} + P_{10} + 6P_{01} + 6P_{11} + P_{02} + P_{12}}{16} & \quad \frac{P_{00} + P_{20} + 6P_{01} + 6P_{11} + 36P_{11} + 6P_{21} + 6P_{12} + P_{02} + P_{22}}{64} \\
\frac{P_{00} + P_{01} + P_{10} + P_{11}}{4} & \quad \frac{P_{00} + P_{01} + 6P_{10} + 6P_{11} + P_{20} + P_{21}}{16} \\
\frac{P_{00} + P_{20} + 6P_{01} + 6P_{10} + 36P_{11} + 6P_{21} + 6P_{12} + P_{02} + P_{22}}{64}
\end{align*}
\]  

(b) Second round of centroid averaging

**Figure 18:** The control points for one level of subdivision of a uniform bicubic B-spline surface constructed by applying two rounds of centroid averaging to the control points in Figure 4.

As expected Figure 18 illustrates three kinds of stencils:

\[Q_F = \frac{P_{00} + P_{01} + P_{10} + P_{11}}{4}\]  

(3.3)

\[Q_E = \frac{P_{00} + P_{10} + 6P_{01} + 6P_{11} + P_{02} + P_{12}}{16}\]  

(3.4)

\[Q_E = \frac{P_{00} + P_{01} + 6P_{10} + 6P_{11} + P_{20} + P_{21}}{16}\]  

(3.5)

Equation (3.3) represents the face stencil; Equation (3.4) represents the two edge stencils, one for vertical edges and one for horizontal edges; and Equation (3.5) represents the vertex stencil. We illustrate these stencils schematically in Figure 19.

**Figure 19:** Stencils for subdivision of bicubic B-spline surfaces: (a) face stencil, (b) edge stencil, and (c) vertex stencil. The position of the new control point (clear diamond) is computed by multiplying the original control points (black discs) by the associated fractions and summing the results. Note that there are two edge stencils: one for horizontal edges and one for vertical edges. Only the horizontal edge stencil is illustrated here, since the vertical edge stencil is much the same, except that the fractions 6/16 lie along horizontal edges instead of along vertical edges.
3.2.2 Stencils for Extraordinary Vertices. To extend stencils from uniform bicubic B-spline surfaces to subdivision surfaces built from arbitrary quadrilateral meshes, we need not alter the face stencil or the edge stencil; all we need to do is to generalize the vertex stencil to extraordinary vertices. This vertex stencil should be affine invariant (the fractions should sum to one), symmetric with respect to the surrounding vertices, and reduce to the vertex stencil for bicubic B-spline surfaces for ordinary vertices with valence four. Also, we want the stencil to generate smooth surfaces -- that is, surfaces with at least one continuous derivative -- at the limits of the extraordinary vertices. Two such stencils are provided in Figure 20; both satisfy all of our constraints, but the Catmull-Clark stencil tends to generate more rounded surfaces.

![Vertex Stencils](image)

*Figure 20:* Two vertex stencils for extraordinary vertices with valence $n$: (a) a simple vertex stencil and (b) the Catmull-Clark stencil. Notice that both stencils reduce to the vertex stencil for bicubic B-splines at ordinary vertices, i.e. vertices where the valence $n = 4$ (see Figure 19(c)).

4. Triangular Meshes

A triangular mesh is a collection of triangles, where each pair of triangles are either disjoint or share a common edge and each edge belongs to at most two triangles. Thus a triangular mesh is much like a quadrilateral mesh, except that the faces are triangles instead of quadrilaterals. A rectangular array of control points generates a regular quadrilateral mesh. Splitting each face in this quadrilateral mesh along a fixed diagonal direction generates a regular triangular mesh, a mesh where each interior vertex is adjacent to six vertices, edges, and faces. The control points for the three direction box splines form a triangular mesh because the vector (1,1) splits each quadrilateral in the mesh generated by the control points into a pair of triangles. Therefore three direction box splines are examples of surfaces generated by subdivision starting from a triangular mesh.

Triangular meshes are quite common in Geometric Modeling, perhaps even more common than quadrilateral meshes. Unlike the control points for three direction box splines, the vertices of an arbitrary triangular mesh are not constrained to be generated from a rectangular array; all that is
required is that the vertices can be joined into non-overlapping triangular faces, where each pair of faces are either disjoint or share a common edge and each edge belongs to at most two faces. For example, the faces of an octahedron form a triangular mesh, even though the vertices of the octahedron do not form a regular mesh, since in the octahedron each vertex has only four adjacent faces (see Figure 24). Two additional examples of triangular meshes that are not regular are illustrated in Figure 23(a) and Figure 25.

The goal of this section is to construct smooth surfaces via subdivision starting from triangular meshes of arbitrary topology. We shall explore the same two paradigms for subdivision of triangular meshes that we investigated for quadrilateral meshes: centroid averaging and stencils.

4.1 Centroid Averaging for Triangular Meshes. Three direction quartic box spline surfaces play the same basic role for subdivision algorithms of triangular meshes that uniform bicubic tensor product B-spline surfaces play for subdivision algorithms of quadrilateral meshes. Therefore we shall begin our investigation of subdivision algorithms for triangular meshes by taking another look at the subdivision algorithm for three direction quartic box splines.

4.1.1 Three Direction Quartic Box Splines. Much like the subdivision algorithm for uniform bicubic B-spline surfaces, the subdivision algorithm for three direction quartic box spline surfaces can be separated into two distinct stages: topology and geometry. In Section 2, we built the subdivision algorithm for one level of the three direction quartic box spline by averaging the control points generated by one level of the subdivision algorithm for the three direction piecewise linear box spline along the vectors \((1,0),(0,1),(1,1)\). Computing the control points for the three direction piecewise linear box spline is equivalent to changing the topology of the mesh; averaging these control points along the vectors \((1,0),(0,1),(1,1)\) is equivalent to altering the geometry of the mesh.

In the topological stage, we keep the original control points and we insert new control points at the midpoints of the edges of the triangular mesh (see Figure 6 and Figure 22(b)). Thus we alter the connectivity of the mesh, adding new edges and faces by connecting the midpoints of adjacent edges. Each edge is split into two edges and each face is split into four faces, but as usual in the topological phase we do not alter the shape of the mesh.

In the geometric stage, we reposition these control points by averaging the control points along the vectors \((1,0),(0,1),(1,1)\). Thus in the geometric stage we change the shape of the mesh, but we do not alter the connectivity of the vertices, edges, and faces (see Figure 22(c)).

There is, however, another somewhat easier way to reposition the vertices. We can generate the same points by applying the stencil in Figure 21 to the control points for the three direction piecewise linear box spline. This result is easy to verify: simply overlay the stencil in Figure 21 on
top of the points in Figure 6 and multiply. As the stencil in Figure 21 moves through the array of control points in Figure 6, these products generate the points in Figure 7 (see Exercise 4).

![Figure 21](image)

**Figure 21.** A stencil for generating the control points for one level of subdivision for the three direction quartic box spline from the control points for one level of subdivision of the three direction piecewise linear box spline. The point (clear diamond) at the center is repositioned by multiplying the points (black discs and clear diamond) by adjacent fractions and adding the results.

Serendipitously, the stencil in Figure 21 can be decomposed into the average of six simpler stencils. Indeed if we represent each stencil as a $3 \times 3$ matrix, then we have the following identity:

$$
\begin{pmatrix}
0 & 1 & 1 \\
1 & 2 & 1 \\
1 & 1 & 0
\end{pmatrix}
= \frac{1}{6}
\begin{pmatrix}
0 & 3 & 3 \\
0 & 8 & 8 \\
0 & 8 & 0
\end{pmatrix}
+ \frac{1}{6}
\begin{pmatrix}
0 & 3 & 0 \\
0 & 8 & 8 \\
0 & 8 & 0
\end{pmatrix}
+ \frac{1}{6}
\begin{pmatrix}
0 & 0 & 3 \\
0 & 8 & 8 \\
0 & 8 & 0
\end{pmatrix}

+ \frac{1}{6}
\begin{pmatrix}
0 & 0 & 0 \\
0 & 3 & 2 \\
0 & 3 & 0
\end{pmatrix}
+ \frac{1}{6}
\begin{pmatrix}
0 & 0 & 0 \\
0 & 3 & 2 \\
0 & 3 & 0
\end{pmatrix}
+ \frac{1}{6}
\begin{pmatrix}
0 & 0 & 0 \\
0 & 3 & 2 \\
0 & 3 & 0
\end{pmatrix}
\tag{4.1}
$$

Equation (4.1) has a geometric interpretation. Consider any interior vertex $v$ in the triangular mesh generated by the subdivision algorithm for the three direction piecewise linear box spline. Each of the six matrices on the right hand side of Equation (4.1) represents a stencil for one of the six triangles in the mesh with $v$ as a vertex. To reposition $v$ for the three direction quartic box spline, take the weighted centroid of each of these triangles with $v$ assigned the weight $2/8$ and the other two vertices assigned the weights $3/8$. Then take the centroid of these weighted centroids (see Figure 22(c)). Note that unlike centroid averaging for bicubic B-splines, the triangle centroids are weighted centroids, so we cannot reuse the same triangle centroids to reposition different vertices; rather we must compute different weighted centroids for each triangle in order to reposition different vertices. Despite this anomaly, the advantage of this somewhat convoluted approach is that we can easily extend this technique to arbitrary triangular meshes, even to triangular meshes with arbitrary topology.
Figure 22: One level of subdivision for the three direction quartic box spline. In (a) we illustrate two faces of the original mesh. In the topological stage (b), new control points (yellow) are inserted at the midpoints of the edges, and new edges and faces are added to the mesh by connecting vertices along adjacent edges. Thus each edge is split into two edges and each face is split into four faces. In the geometric stage (c), each control point -- yellow or black -- is repositioned to the centroid (blue) of the weighted centroids (green) of the faces adjacent to the control point, but the topology of the mesh is not changed. Note that since the green centroids are weighted centroids, we cannot reuse the same centroids to reposition different control points, rather we must recompute these centroids for each vertex. Compare to Figure 12 for bicubic B-splines.

4.1.2 Arbi-trary Triangular Meshes. As with three direction quartic box splines, there are two main phases to centroid averaging for arbitrary triangular meshes: topology and geometry.

To refine the topology of the mesh, we proceed exactly as we did for three direction quartic box splines: we keep the original vertices and we insert new edge vertices at the midpoints of the edges. Thus each edge is split into two edges. We also introduce new edges to connect the midpoints of adjacent edges and new faces surrounded by these new edges, splitting each of the original faces into four faces (see Figure 23(b)).

To alter the geometry of the mesh, we reposition each vertex of the mesh to the centroid of the weighted centroids of the adjacent faces. The only difference between subdivision for three direction quartic box splines and subdivision for arbitrary triangular meshes is that for arbitrary triangular meshes there need not be six faces adjacent to each interior vertex. For example, in the octahedron there are only four faces adjacent to each of the vertices. An interior vertex of a triangular mesh where the number of adjacent faces, edges, or vertices is not equal to six is called an extraordinary vertex. Again, the number of faces, edges, or vertices adjacent to a vertex is called the valence of the vertex. We can reposition each vertex $Q$ in an arbitrary triangular mesh by the following rule: Take the weighted centroid of each of the triangles containing $Q$ with $Q$ assigned the weight $2/8$ and the other two vertices assigned the weights $3/8$; then take the centroid of these weighted centroids. Thus each vertex $Q$ of the mesh is repositioned by the formula

$$Q \rightarrow \frac{1}{n} \sum_{k=1}^{n} C_k,$$

where $C_1, \ldots, C_n$ are the weighted centroids of the faces adjacent to $Q$ and $n$ is the valence of $Q.$
After several levels of subdivision, most of the interior vertices of a triangular mesh are ordinary vertices and the extraordinary vertices become more and more isolated because during the topological phase, each face is subdivided into four faces by the edges joining the midpoints of the surrounding edges. Thus the new edge vertices and the new face vertices all have valence six (see Figure 23(b)), so all the new vertices introduced by subdivision are ordinary vertices. Also the valence of each of the original vertices is unchanged, since the number of faces surrounding a vertex does not change during subdivision. These results are analogous to similar results for quadrilateral meshes (see Section 3.1.2). Hence during subdivision ordinary vertices remain ordinary vertices and extraordinary vertices remain extraordinary vertices with the same valence (see Figure 23(b)). Since all the new vertices have valence six, locally the new triangular mesh looks exactly like a triangular mesh for a three direction quartic box spline surface; only vertices from the original mesh that did not have valence six do not have six adjacent faces. Therefore, just like three direction quartic box spline surfaces, the surfaces generated in the limit by iterating centroid averaging for triangular meshes have two continuous derivatives everywhere; the only exceptions are at the limits of extraordinary vertices. At the limits of extraordinary vertices these surfaces have only one continuous derivative, except at the limits of extraordinary vertices of valence three where these limit surfaces are continuous but not necessarily differentiable (see Warren and Weimer, Chapter 8).

![Figure 23](image-url)

**Figure 23:** One level of centroid averaging. In (a) we illustrate eight faces of a triangular mesh surrounding an extraordinary vertex (blue). In the topological stage (b), new vertices (yellow) are inserted at the midpoints of the edges of the triangular mesh, and new edges and faces are added to the mesh by joining the midpoints of adjacent edges. In the geometric stage (c), each vertex -- blue, yellow or black -- is repositioned to the centroid of the weighted centroids (green) of the faces adjacent to the vertex, but the topology of the mesh is not changed. Notice that all the new vertices (yellow) are ordinary vertices with valence six. Also the extraordinary vertex (blue) at the center with valence eight remains an extraordinary vertex with valence eight and is repositioned to the centroid of eight adjacent weighted centroids (green). Since the green centroids are weighted centroids, we cannot reuse the same centroids to reposition different vertices; rather we must recompute these green centroids for each vertex. Compare to Figure 13 for quadrilateral meshes.
Figure 24: Three levels of centroid averaging applied to an octahedron (left). Notice the extraordinary vertices of valence four at the corners of the octahedron.

Figure 25: Centroid averaging applied to two tetrahedra meeting at a vertex. The original triangular mesh is on the left; the first three levels of subdivision are illustrated successively on the right.

4.2 Stencils for Triangular Meshes. For bicubic B-spline surfaces and more generally for quadrilateral meshes of arbitrary topology, we have seen that stencils provide an alternative to centroid averaging for generating smooth surfaces via subdivision. Centroid averaging separates the computation of new vertices during subdivision into two phases: topology and geometry. After the insertion of new vertices in the topological phase, all the vertices of the mesh are repositioned in the geometric phase by centroid averaging. But centroid averaging is a bit cumbersome for three direction quartic box splines and more generally for triangular meshes of arbitrary topology because the centroids that must be averaged are weighted centroids. Thus we need to compute a different weighted centroid for each triangle in order to reposition each vertex. In contrast, stencils allow us to compute the positions of all the new vertices in terms of the positions of the original
vertices without resorting to a different stencil for each vertex. For regular triangular meshes all we need are a single edge stencil and a single vertex stencil. For triangular meshes of arbitrary topology, we shall also require additional stencils for extraordinary vertices. Thus for three direction quartic box splines and more generally for triangular meshes of arbitrary topology, stencils may be easier to implement than centroid averaging.

4.2.1 Stencils for Three Direction Quartic Box Splines. For three direction quartic box splines Figure 7 provides essentially two distinct explicit formulas for computing the new control points from the original control points after one level of subdivision:

\[
Q_E = \frac{P_{00} + 3P_{01} + 3P_{11} + P_{12}}{8}
\]

\[
Q_E = \frac{P_{00} + 3P_{10} + 3P_{11} + P_{21}}{8}
\]

\[
Q_E = \frac{3P_{10} + P_{11} + P_{20} + 3P_{21}}{8}
\]

\[
Q_V = \frac{P_{00} + P_{01} + 10P_{11} + P_{12} + P_{21} + P_{22}}{16}
\]

Equation (4.2) represents the three edge stencils: one for horizontal, one for vertical, and one for diagonal edges. Notice that all three formulas are essentially the same. Equation (4.3) represents the vertex stencil. We illustrate these two stencils schematically in Figure 26.

![Figure 26: Stencils for subdivision of three direction quartic box splines: (a) the edge stencil and (b) the vertex stencil. The position of the new control point (clear diamond) is computed by multiplying the original control points (black discs) by the associated fractions and summing the results. There are three edge stencils, but they are all essentially the same so only the vertical edge stencil is shown here.](image)

4.2.2 Stencils for Extraordinary Vertices. To extend stencils from three direction quartic box splines to subdivision surfaces built from arbitrary triangular meshes, we need not alter the edge
stencil; all we need to do is to generalize the vertex stencil to extraordinary vertices. Similar to vertex stencils at extraordinary vertices of quadrilateral meshes, this vertex stencil should be affine invariant (the fractions should sum to one), symmetric with respect to surrounding vertices, and reduce to the stencil for three direction quartic box splines for vertices with valence six. Also, we want the stencil to generate smooth surfaces -- that is, surfaces with at least one continuous derivative -- at the limits of the extraordinary vertices. Two such stencils are provided in Figure 27. Both stencils satisfy all of our constraints, but only the Loop stencil is guaranteed to generate surfaces with bounded curvatures.

![Figure 27](image)

**Figure 27:** Two vertex stencils for extraordinary vertices with valence \( n \): (a) a simple vertex stencil and (b) the Loop stencil. For the Loop stencil

\[
w(n) = \frac{5}{8} - \left( \frac{3}{8} + \frac{1}{4} \cos \left( \frac{2\pi}{n} \right) \right)^2.
\]

Notice that both stencils reduce to the vertex stencil for three direction quartic box splines at ordinary vertices, i.e. vertices where the valence \( n = 6 \) (see Figure 26(b)).

5. **Summary**

Subdivision is an important tool in Geometric Modeling for several reasons:

i. Subdivision algorithms allow us to replace complicated mathematical formulas with simple computational procedures.

ii. Subdivision algorithms are easy to understand and simple to implement.

iii. Subdivision algorithms can generate a large class of smooth functions.

iv. Subdivision algorithms on polyhedral meshes can produce smooth shapes with arbitrary topology.

Thus subdivision provides a simple approach to generating a wide variety of smooth curves and surfaces.
This lecture presents three distinct paradigms for generating subdivision surfaces: split and average (box splines), centroid averaging (meshes of arbitrary topology) and stencils (vertex, edge and face stencils as well as special stencils for extraordinary vertices). Box spline surfaces are generalizations of uniform tensor product B-spline surfaces, constructed by a classical split and average approach to subdivision starting from a rectangular array of control points. These box spline surfaces are important because they provide a paradigm for constructing subdivision algorithms for triangular meshes of arbitrary topology. Indeed subdivision surfaces built by centroid averaging or by stencils starting from quadrilateral or triangular meshes of arbitrary topology are generalizations of standard subdivision procedures for bicubic tensor product B-splines and three direction quartic box splines. One of the big advantages of subdivision algorithms for polyhedral meshes is that unlike tensor product B-spline or box spline schemes, subdivision procedures for polyhedral meshes can generate smooth surfaces with arbitrary topology.

An important leitmotif of this chapter is that subdivision procedures can be partitioned into two separate phases: refining the topology and altering the geometry -- that is, changing connectivity and modifying shape. For box splines, we refine the topology by splitting each control point into four control points, leaving the position of each control point unchanged. For polyhedral meshes, we refine the topology by inserting new vertices and edges, and splitting each face into four faces, leaving the positions of the original vertices unchanged. In the geometric phase for box splines, we alter the positions of the control points by averaging these points along prespecified directions in the plane of their integer indices. In the geometric phase for polyhedral meshes, we change the positions of the vertices by taking the centroid of the centroids of adjacent faces. Stencils allow us to combine refining the topology and modifying the geometry of a polyhedral mesh into a single phase.

Below we summarize for easy access and comparison our three approaches to subdivision algorithms -- split and average, centroid averaging, and stencils -- for bicubic tensor product B-splines and three direction quartic box splines, surfaces defined over regular meshes whose subdivision algorithms serve as models for subdivision procedures on quadrilateral and triangular meshes of arbitrary topology. We also review centroid averaging for quadrilateral and triangular meshes. Finally we recall the stencils for extraordinary vertices in quadrilateral and triangular meshes.
5.1 Bicubic Tensor Product B-Splines and Three Direction Quartic Box Splines. For bicubic tensor product B-splines and three direction quartic box splines there are three approaches to subdivision algorithms: split and average, centroid averaging, and stencils.

5.1.1 Split and Average

\[ P_{2i,2}^{0,0} = P_{2i+1,2}^{0,0} = P_{2i,2}^{0,0} = P_{2i+1,2}^{0,0} = P_{i,j} \]  

(see Figure 28)

**Average -- Bicubic B-Splines**

\[ P_{i,j}^{k,0} = \frac{P_{i,j}^{k-1,0} + P_{i+1,j}^{k-1,0}}{2}, \quad k = 1, 2, 3 \]  

(horizontal direction)

\[ P_{i,j}^{3,k} = \frac{P_{i,j}^{3,k-1} + P_{i,j+1}^{3,k-1}}{2}, \quad k = 1, 2, 3 \]  

(vertical direction)

**Average -- Three Direction Quartic Box Splines**

\[ P_{i,j}^{1,0} = \frac{P_{i,j}^{0,0} + P_{i+1,j}^{0,0}}{2} \]  

(horizontal direction)

\[ P_{i,j}^{1,1} = \frac{P_{i,j}^{1,0} + P_{i,j+1}^{1,0}}{2} \]  

(vertical direction)

\[ P_{i,j}^{k+1,k+1} = \frac{P_{i,j}^{k,k} + P_{i+1,j+1}^{k,k}}{2}, \quad k = 1, 2 \]  

(diagonal direction)

\[ \vdots \]

\[ P_{00} \quad P_{01} \quad P_{11} \quad \cdots \]

\[ P_{01} \quad P_{01} \quad P_{11} \quad \cdots \]

\[ P_{00} \quad P_{00} \quad P_{10} \quad P_{10} \quad \cdots \]

\[ P_{00} \quad P_{00} \quad P_{10} \quad P_{10} \quad \cdots \]

**Figure 28:** Splitting each control point into four points.
5.1.2 Centroid Averaging

**Bicubic B-Splines**

- **Refine the Topology**

\[
P_{2i,2j}^1 = P_{i,j}
\]

\[
P_{2i+1,2j}^1 = \frac{P_{i,j} + P_{i+1,j}}{2}
\]

\[
P_{2i,2j+1}^1 = \frac{P_{i,j} + P_{i,j+1}}{2}
\]

\[
P_{2i+1,2j+1}^1 = \frac{P_{i,j} + P_{i+1,j} + P_{i,j+1} + P_{i+1,j+1}}{4}
\]

- **Modify the Geometry**

\[
R_{i,j}^k = \frac{P_{i,j}^k - 1 + P_{i+1,j}^k - 1 + P_{i,j+1}^k - 1 + P_{i+1,j+1}^k - 1}{4}
\]

\[k = 2, 3\]

**Three Direction Quartic Box Splines**

- **Refine the Topology** (Insert New Edge Vertices)

\[
P_{2i,2j}^1 = P_{i,j}
\]

\[
P_{2i+1,2j}^1 = \frac{P_{i,j} + P_{i+1,j}}{2}
\]

\[
P_{2i,2j+1}^1 = \frac{P_{i,j} + P_{i,j+1}}{2}
\]

\[
P_{2i+1,2j+1}^1 = \frac{P_{i,j} + P_{i+1,j+1}}{2}
\]

- **Modify the Geometry** (Weighted Centroid Averaging)

\[
R_{i,j}^k = \frac{2P_{i,j}^1 + 3Q_k + 3Q_{k+1}}{8}, \text{ where } Q_1, \ldots, Q_6 \text{ are the six vertices adjacent to } P_{ij}
\]

\[
R_{i,j}^2 = \frac{P_{i,j+1}^2 + P_{i,j-1}^2 + P_{i,j-3}^2 + P_{i,j+3}^2 + P_{i,j+5}^2 + P_{i,j-5}^2}{6}
\]

\[
R_{i,j}^3 = \frac{P_{i,j+1}^3 + P_{i,j-1}^3 + P_{i,j-4}^3 + P_{i,j+4}^3 + P_{i,j+6}^3 + P_{i,j-6}^3}{6}
\]
Figure 30: Centroid Averaging -- Three Direction Quartic Box Splines. First (b) refine the topology by inserting a new edge vertex (yellow) at the center of each edge, and then connecting vertices on adjacent edges to split each face into four faces. The new control point (blue) in (c) is repositioned to the centroid of the weighted centroids (green) of the adjacent faces.

5.1.3 Stencils. Stencils represent explicit expressions in terms of the original control points for the new control points after one level of subdivision. For bicubic B-splines these formulas come in three basic forms: expressions for face points, expressions for edge points and expressions for vertex points. For three direction quartic box splines, these formulas come in only two basic forms: expressions for edge points and expressions for vertex points.

**Bicubic B-Splines**

\[
Q_{\text{Face}} = \frac{P_{00} + P_{01} + P_{10} + P_{11}}{4}
\]

\[
Q_{\text{Edge}} = \frac{P_{00} + P_{10} + 6P_{01} + 6P_{11} + P_{02} + P_{12}}{16} \quad \text{or} \quad \frac{P_{00} + P_{01} + 6P_{10} + 6P_{11} + P_{20} + P_{21}}{16}
\]

\[
Q_{\text{Vertex}} = \frac{P_{00} + P_{20} + 6P_{01} + 6P_{10} + 36P_{11} + 6P_{21} + 6P_{12} + P_{02} + P_{22}}{64}
\]

Figure 31: Stencils for bicubic B-splines: (a) face stencil, (b) edge stencil, and (c) vertex stencil.

**Three Direction Quartic Box Splines**

\[
Q_{\text{Edge}} = \frac{P_{00} + 3P_{01} + 3P_{11} + P_{12}}{8} \quad \text{or} \quad \frac{P_{00} + 3P_{10} + 3P_{11} + P_{21}}{8} \quad \text{or} \quad \frac{3P_{10} + P_{11} + P_{20} + 3P_{21}}{8}
\]

\[
Q_{\text{Vertex}} = \frac{P_{00} + P_{01} + P_{10} + 10P_{11} + P_{12} + P_{21} + P_{22}}{16}
\]
Figure 32: Stencils for Three Direction Quartic Box Splines: (a) the edge stencil and (b) the vertex stencil.

5.2 Centroid Averaging for Meshes of Arbitrary Topology

Figure 33: Quadrilateral Meshes. Each vertex is repositioned to (c) the centroid of the centroids (blue) of the adjacent faces after (b) refining the topology by splitting each quadrilateral face into four quadrilateral faces.

Figure 34: Triangular Meshes. Each vertex is repositioned to (c) the centroid of the weighted centroids (green) of the adjacent faces after (b) refining the topology by splitting each triangular face into four triangular faces.
5.3 Stencils for Extraordinary Vertices. For quadrilateral meshes of arbitrary topology, the edge and face stencils are the same as the edge and face stencils for tensor product bicubic B-splines. Similarly, for triangular meshes, the edge stencils are the same as the edge stencils for three direction quartic box splines. Also for quadrilateral meshes, the vertex stencil at regular vertices (vertices with valence four) is the same as the vertex stencil for tensor product bicubic B-splines, and for triangular meshes the vertex stencil at regular vertices (vertices with valence six) is the same as the vertex stencil for three direction quartic box splines. Therefore for meshes of arbitrary topology, we need only specify the vertex stencils at extraordinary vertices.

![Stencils for Extraordinary Vertices](image)

(a) simple vertex stencil  
(b) Catmull-Clark stencil

**Figure 35:** Quadrilateral Meshes. Two vertex stencils for extraordinary vertices with valence $n$: (a) a simple vertex stencil and (b) the Catmull-Clark stencil.

![Stencils for Extraordinary Vertices](image)

(a) simple vertex stencil  
(b) Loop stencil

**Figure 36:** Triangular Meshes. Two vertex stencils for extraordinary vertices with valence $n$: (a) a simple vertex stencil and (b) the Loop stencil. For the Loop stencil

\[
w(n) = \frac{5}{8} \left( \frac{3}{8} + \frac{1}{4} \cos \left( \frac{2\pi}{n} \right) \right)^2.
\]
Exercises:

1. Let \( \{P_{ij}\} \) be a rectangular array of control points.
   
a. Show that averaging adjacent control points in the \( s \)-direction and then averaging the resulting adjacent control points in the \( t \)-direction is equivalent to first averaging adjacent control point in the \( t \)-direction and then averaging the resulting adjacent control points in the \( s \)-direction by showing that in both cases the new control points are given by
   \[
   Q_{i,j} = \frac{P_{i,j} + P_{i+1,j} + P_{i,j+1} + P_{i+1,j+1}}{4}.
   \]

   b. Generalize the result in part a by showing that averaging \( v \)-adjacent points and then averaging \( w \)-adjacent points is equivalent to first averaging \( w \)-adjacent points and then averaging \( v \)-adjacent points by showing that in both cases the new control points are given by
   \[
   Q_{i,j} = \frac{P_{i,j} + P_{i+v_1,j+v_2} + P_{i+w_1,j+w_2} + P_{i+v_1+w_1,j+v_2+w_2}}{4}.
   \]

2. Consider the box spline surface generated by the control points \( \{P_{ij}\} \) and the vectors \( V = \{(1,0),(1,0),(0,1),(0,1),(1,1)\} \).
   
a. What is the degree of this surface?
   b. How smooth is this surface?
   c. Show that the control points for this surface after one level of subdivision are given by the array of points in Figure 37.
   d. Using the result in part c, derive the control points for three direction quartic box splines.

   \[
   \begin{align*}
   &5P_{01} + P_{11} + P_{02} + P_{12} & & 8 & & 5P_{11} + P_{21} + P_{12} + P_{22} & & 8 \\
   &P_{00} + 2P_{01} + P_{11} & & 4 & & 5P_{10} + P_{11} + 5P_{12} & & 4 \\
   &P_{00} + P_{10} + P_{01} + 5P_{11} & & 8 & & P_{10} + 2P_{11} + P_{21} & & 4 \\
   &5P_{00} + P_{10} + P_{01} + P_{11} & & 8 & & 5P_{10} + P_{20} + P_{11} + P_{21} & & 8
   \end{align*}
   \]

   Figure 37: The control points after one level of subdivision for the 3-direction box spline in Exercise 2.
3. Consider the box spline surface generated by the control points \( \{P_{ij}\} \) and the vectors \( V = \{(1,0),(0,1), (1,1), (-1,1)\} \).
   a. What is the degree of this surface?
   b. How smooth is this surface?
   c. Show that the control points for this surface after one level of subdivision are given by the array of points in Figure 38.

\[
\begin{align*}
\frac{P_{02} + 2P_{01} + P_{11}}{4} & \quad \frac{P_{01} + 2P_{11} + P_{12}}{4} & \quad \frac{P_{12} + 2P_{11} + P_{21}}{4} \\
\frac{P_{00} + 2P_{01} + P_{11}}{4} & \quad \frac{P_{01} + 2P_{11} + P_{10}}{4} & \quad \frac{P_{10} + 2P_{11} + P_{21}}{4} \\
\frac{P_{01} + 2P_{00} + P_{10}}{4} & \quad \frac{P_{00} + 2P_{10} + P_{11}}{4} & \quad \frac{P_{11} + 2P_{10} + P_{20}}{4}
\end{align*}
\]

Figure 38: The control points after one level of subdivision for the 4-direction box spline in Exercise 3.

A stencil \( S \) is called a universal stencil for a subdivision algorithm if all the control points -- face points, edge points, and vertex points -- after one level of subdivision can be generated by sliding the stencil \( S \) over an array of points \( Q \).

4. Show that the stencil in Figure 21 is a universal stencil for the three direction quartic box spline by verifying that sliding this stencil on the array of control points in Figure 6 for one level of subdivision of the three direction linear box spline generates the control points in Figure 7 for one level of subdivision for the three direction quartic box spline.

5. Consider the stencil in Figure 39.
   a. Show that this stencil is a universal stencil for the uniform tensor product bicubic B-splines by verifying that sliding this stencil on the array of control points in Figure 4 for one level of subdivision for bilinear B-splines generates the control points in Figure 18(b) for one level of subdivision for bicubic B-splines.
   b. Verify that if we represent this stencil as a \( 3 \times 3 \) matrix, then we have the following identity:

\[
\begin{pmatrix}
\frac{1}{16} & \frac{2}{16} & \frac{1}{16} \\
\frac{2}{16} & \frac{4}{16} & \frac{2}{16} \\
\frac{1}{16} & \frac{2}{16} & \frac{1}{16}
\end{pmatrix} = \frac{1}{4}\begin{pmatrix}
\frac{1}{4} & \frac{1}{4} & 0 \\
\frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\
0 & 0 & 0
\end{pmatrix} + \frac{1}{4}\begin{pmatrix}
0 & \frac{1}{4} & \frac{1}{4} \\
\frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\
0 & 0 & 0
\end{pmatrix} + \frac{1}{4}\begin{pmatrix}
0 & 0 & 0 \\
\frac{1}{4} & \frac{1}{4} & 0 \\
0 & \frac{1}{4} & \frac{1}{4}
\end{pmatrix} + \frac{1}{4}\begin{pmatrix}
0 & 0 & 0 \\
\frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\
0 & 0 & \frac{1}{4}
\end{pmatrix}
\]
c. Conclude from part b that after bilinear averaging, each control point in the Lane-Riesenfeld algorithm for a bicubic B-spline is repositioned to the centroid of the centroids of the faces adjacent to the control point.

\[
\begin{array}{cccc}
\frac{1}{16} & \frac{2}{16} & \frac{1}{16} \\
\frac{2}{16} & \frac{4}{16} & \frac{2}{16} \\
\frac{1}{16} & \frac{2}{16} & \frac{1}{16}
\end{array}
\]

**Figure 39.** A universal stencil for generating the control points for one level of subdivision for bicubic B-splines from the control points for one level of subdivision for bilinear B-splines. The point (clear diamond) at the center is repositioned by multiplying the points (black discs and clear diamond) by the adjacent fractions and adding the results.

6. Consider the stencil in Figure 40.
   a. Show that this stencil is a universal stencil for cubic B-spline curves by verifying that sliding this stencil along the array of control points in Figure 1 for one level of subdivision for the linear B-splines generates the control points in Figure 16 for one level of subdivision for the cubic B-splines.
   b. Show that this stencil can be decomposed into the average of two simpler stencils.
   c. Conclude from part b that after linear averaging, each control point in the Lane-Riesenfeld algorithm for a cubic B-spline curve is repositioned to the midpoint of the midpoints of the edges adjacent to the control point.

\[
\frac{1}{4} \quad \frac{2}{4} \quad \frac{1}{4}
\]

**Figure 40.** A universal stencil for generating the control points for one level of subdivision for cubic B-spline curves from the control points for one level of subdivision for linear B-spline curves. The point (clear diamond) at the center is repositioned by multiplying the points (black discs and clear diamond) by the adjacent fractions and adding the results.

7. Show that the stencil represented by the row vector \((1, 3, 3, 1)\) is a universal stencil for cubic B-spline curves by verifying that sliding this stencil along the array where each of the original control points are doubled generates the control points for one level of subdivision for cubic B-spline curves. Generalize this result to B-spline curves of arbitrary degrees.
8. **The Four Point Scheme.** Let \( \{ P_i \} \) be a collection of control points, and let \( \{ P_i^3 \} \) be the output of one level of the Lane-Riesenfeld algorithm for cubic B-spline curves. Define one level of subdivision for the four point scheme by setting:
\[
P_i^4 = \frac{-P_{i-1}^3 + 4P_i^3 - P_{i+1}^3}{2}.
\]

a. Show that
\[
P_{2i}^4 = P_i \quad \text{(interpolation)}
\]
\[
P_{2i+1}^4 = \frac{-P_{i-1}^3 + 9P_i^3 + 9P_{i+1}^3 - P_{i+2}^3}{16} \quad \text{(four point rule)}
\]

b. Describe the edge stencil and vertex stencil for the four point scheme.

c. Find two distinct universal stencils for the four point scheme.

d. Using the result of part \( a \), conclude that the control polygons generated by iterating the four point scheme converge in the limit to curves that interpolates the original control points.

e. Implement the four point scheme. Observe that the control polygons generated by this subdivision algorithm converge in the limit to smooth curves.

9. Verify that the Loop stencil in Figure 27(b) reduces to the vertex stencil for the three direction quartic box spline in Figure 26(b) when the valence \( n = 6 \).

**Programming Projects:**

1. **Box Splines**
   Implement subdivision for box spline surfaces in your favorite programming language using your favorite API.
   a. Render bicubic B-splines and four direction quartic box splines, using your favorite shading algorithm and hidden surface procedure.
   b. Build some interesting freeform shapes using box spline surfaces.

2. **Quadrilateral Meshes**
   Implement subdivision algorithms for quadrilateral meshes in your favorite programming language using your favorite API.
   a. Include both centroid averaging and Catmull-Clark stencils at extraordinary vertices.
   b. Render these subdivision surfaces, using your favorite shading algorithm and hidden surface procedure.
   c. Build some interesting freeform shapes of arbitrary topology, using subdivision algorithms for quadrilateral meshes of arbitrary topology.
3. **Triangular Meshes**
   Implement subdivision algorithms for triangular meshes in your favorite programming language using your favorite API.
   a. Include both centroid averaging and Loop stencils at extraordinary vertices.
   b. Render these subdivision surfaces, using your favorite shading algorithm and hidden surface procedure.
   c. Build some interesting freeform shapes of arbitrary topology, using subdivision algorithms for triangular meshes of arbitrary topology.

4. **Split and Average -- Geometric Mean**
   Let \( P_k = (x_k, y_k) \) be a collection of points in the \( xy \)-plane, where \( y_k > 0 \), and consider the following variation of the Lane-Riesenfeld algorithm:
   Replace midpoint averaging for the \( y \)-coordinate by the geometric average -- that is, replace the arithmetic mean \( \frac{y_k + y_{k+1}}{2} \) by the geometric mean \( \sqrt{y_k y_{k+1}} \) -- but use the arithmetic mean \( \frac{x_k + x_{k+1}}{2} \) for averaging the \( x \)-coordinates. Thus the new averaging rule is
   \[
   av(P_k, P_{k+1}) = \left( \frac{x_k + x_{k+1}}{2}, \sqrt{y_k y_{k+1}} \right).
   \]
   a. Implement this new subdivision algorithm.
      i. Are the curves you construct always smooth?
      ii. Compare these new curves to the curves generated by the standard Lane-Riesenfeld algorithm for linear, quadratic, and cubic B-splines with the same control points.
   b. Now replace midpoint averaging in the \( x \)-coordinate by the geometric mean, but use the arithmetic mean in the \( y \)-coordinate.
      i. Are the curves still smooth?
      ii. How are these curves related to the curves generated by the standard Lane-Riesenfeld algorithm for linear, quadratic, and cubic B-splines with the same control points?
      iii. How are these curves related to the curves generated in part a?