The Refinement Rules for Catmull-Clark Solids

Kenneth I. Joy
Ron MacCracken

Computer Graphics Research Laboratory
Department of Computer Science
University of California, Davis

Technical Report CSE-96-1

February 3, 1999

The Refinement Rules for Catmull-Clark Solids

Kenneth I. Joy
Ron MacCracken

Technical Report CSE-96-1

Computer Graphics Research Laboratory
Department of Computer Science
University of California, Davis

Abstract

B-Spline curves and surfaces can be defined by a set of control points and a set of refinement rules. These rules act on the defining mesh of control points to create a new refined mesh. Repeated application of the rules generates a sequence of meshes that converge to the curve or surface. In the surface case, Catmull and Clark extended the refinement rules for the rectangular topological meshes of the B-spline form to meshes that have an arbitrary topological structure. In much the same way, the refinement rules for a trivariate B-spline solid can be extended from those that apply to the regular hexahedral topological lattices of the B-spline form to lattices that have an arbitrary topological structure. We present a uniform development of the refinement rules for trivariate B-spline solids and extend the rules to apply to solid lattices of arbitrary topology.

1. Introduction

In geometric modeling, the techniques based upon the Bézier and B-spline paradigms have been at the forefront for over two decades. The single characteristic that insures their popularity is that the curves, surfaces or solids defined by these methods are uniquely defined by a set of control points in three-dimensional space. These control points completely define the B-spline object: In the univariate case, the control points are arranged in a sequence - a one dimensional array; in the bivariate case they are arranged in a two-dimensional array structure; and in the trivariate case, they are arranged in a three-dimensional array. These array structures are quite significant in that they enable us to define the surface and solid in a tensor product form. One benefit of this is that it allows us to use the algorithms and methods generated for B-spline curves to design the algorithms for B-spline surfaces and solids. This ease of definition and ease of design brought about by manipulating a meaningful set of control points is what has made the Bézier and B-spline methods so useful in geometric modeling.

Unfortunately, the regular topological structures imposed on the control points of a B-spline surface or solid limit the usefulness of the B-spline paradigm. For example, the B-spline surface is a tensor product surface where the control points are arranged in an $n \times m$ array. This array structure imposes a rectangular or quadrilateral structure on the set of control points. Consider the case of a control point mesh that is arranged in a triangular structure - such as one may get from approximating triangles on surfaces. The B-spline $n \times m$ array structure cannot be adapted to represent the triangular structure without creating a degeneracy in the resulting surface. As the solid case is just an extension of the surface case (bivariate to trivariate), the problem clearly exists in this case also - perhaps even to a greater extent.

In the case of surfaces, there have been several proposed solutions to this problem, the most important of which center about methods of defining the surface by subdivision. These methods define a set of rules that refine a defining mesh into a "new" mesh that still represents the surface exactly. By repetitively applying these methods, one can generate a sequence of meshes that converges to the surface. The refinement rules for a B-spline surface are straightforward to define. The idea is to adapt these rules so that they can be used with a mesh of arbitrary topology. With some care, they can be developed so that, if the mesh is quadrilateral, then the surface is a B-spline. The refinement methods were first presented by George Chaikin [2] for curves; Riesenfeld [9] proceeded to show that Chaikin's curves were uniform quadratic B-spline curves; Doo and Sabin [3, 4] extended Chaikin's methods to uniform quadratic B-spline surfaces, and then extended the refinement rules for the quadratic case to meshes of an arbitrary topology; and, Catmull and Clark [1] developed the cubic case. These methods have now come into widespread use in geometric modeling. They have been used for interpolation and fairing [5], approximation [6], multiresolution design [7], and as a preprocessing step in the design process [8]. In this paper, we limit ourselves to the cubic case and define the set of refinement rules for solids specified by a mesh of arbitrary topology.

Our strategy is to build up these refinement methods by successively considering the univariate, bivariate and trivariate cases. We will follow the general methodology given by Catmull and Clark, in that we first examine the refinement for a trivariate B-spline solid defined over a hexahedral mesh. We will then extend these rules to solids defined by a mesh of arbitrary topology. In section 2 we begin this analysis by developing the refinement rules for uniform cubic B-spline curves. In section 3 we utilize these univariate rules to develop refinement rules for uniform cubic B-spline surfaces and then present, in section 4, Catmull and Clark's classic generalization of the surface rules to meshes of arbitrary topology. In section 5, we develop the extension of the refinement rules for uniform B-spline surfaces to rules for the trivariate uniform B-spline solid. Finally in section 6, we generalize these rules to meshes of arbitrary topology. We have attempted to be complete in this paper - developing the rules for the most elementary of modeling primitives - curves –

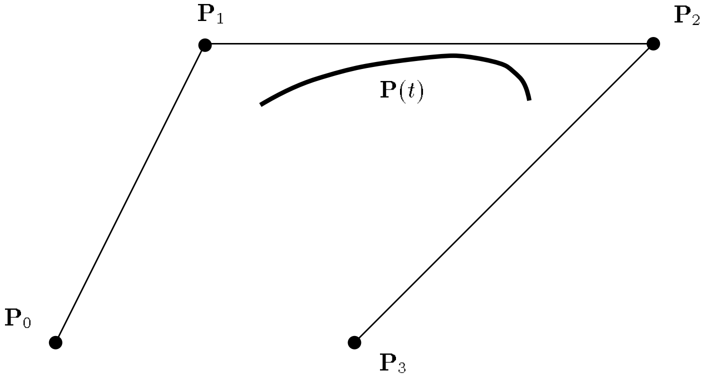
and using these uniformly to develop the rules for higher-level primitives.

2. The Univariate Case

The refinement rules for the univariate case are developed directly from the methods for binary subdivision of the uniform B-spline curve. We first develop the refinement rules for a uniform cubic B-spline curve defined by four control points. We then extend these rules to those work with control polygons of arbitrary length.

2.1. Binary Subdivision

Consider a cubic uniform B-spline curve $\mathbf{P}(t)$ defined by the control polygon consisting of the four points $\mathbf{P}_0$, $\mathbf{P}_1$, $\mathbf{P}_2$ and $\mathbf{P}_3$. Such a curve is shown in the following illustration.



This curve is subdivided into two pieces by applying one of the two splitting matrices

$$S^L = \frac{1}{8}\begin{bmatrix} 4 & 4 & 0 & 0 \\ 1 & 6 & 1 & 0 \\ 0 & 4 & 4 & 0 \\ 0 & 1 & 6 & 1 \end{bmatrix}$$

$$S^R = \frac{1}{8}\begin{bmatrix} 1 & 6 & 1 & 0 \\ 0 & 4 & 4 & 0 \\ 0 & 1 & 6 & 1 \\ 0 & 0 & 4 & 4 \end{bmatrix}$$

to the control polygon. (When applied to the control polygon S^L gives the control points for the first half of the curve, and S^R gives the control points of the second half.) These matrices induce affine operations on the control points, as the sum of the elements of each row of the matrix is one.

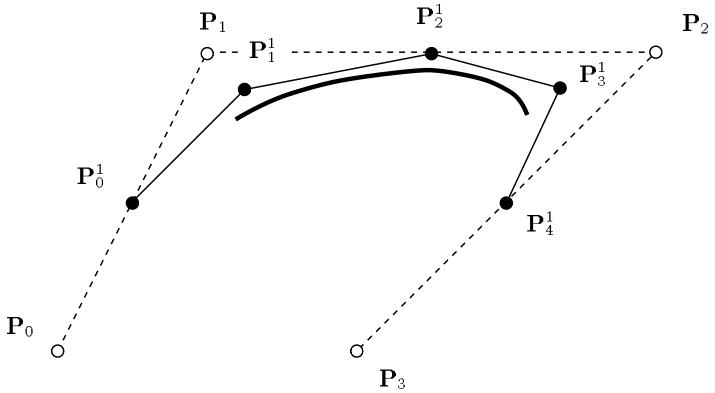
By examining the rows of the two matrices, we can see that five unique points are generated and we can combine the two into a single 5×4 matrix

$$\frac{1}{8}\begin{bmatrix} 4 & 4 & 0 & 0 \\ 1 & 6 & 1 & 0 \\ 0 & 4 & 4 & 0 \\ 0 & 1 & 6 & 1 \\ 0 & 0 & 4 & 4 \end{bmatrix}$$

which can be applied to the control polygon by

$$\begin{bmatrix} \mathbf{P}_0^1 \\ \mathbf{P}_1^1 \\ \mathbf{P}_2^1 \\ \mathbf{P}_3^1 \\ \mathbf{P}_4^1 \end{bmatrix} = \frac{1}{8}\begin{bmatrix} 4 & 4 & 0 & 0 \\ 1 & 6 & 1 & 0 \\ 0 & 4 & 4 & 0 \\ 0 & 1 & 6 & 1 \\ 0 & 0 & 4 & 4 \end{bmatrix} \begin{bmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{bmatrix}$$

generating a new control polygon which serves as the refinement of the original. The five control points of this new control polygon specify the two subdivided halves of the curve ($\mathbf{P}_0^1$, $\mathbf{P}_1^1$, $\mathbf{P}_2^1$, and $\mathbf{P}_3^1$ specify the first half, and $\mathbf{P}_1^1$, $\mathbf{P}_2^1$, $\mathbf{P}_3^1$, and $\mathbf{P}_4^1$ specify the second half), and therefore uniquely specify the curve itself.



We note that three of the new control points appear to lie at the midpoints of the three respective line segments. These points will be classified as "edge points". The other points all lie close to one of the interior vertices ($\mathbf{P}_1$ and $\mathbf{P}_2$) of the original control polygon, and will be called "vertex points".

With this classification, denote the new control polygon generated by the binary subdivision method as

$\{\mathbf{E}_0, \mathbf{V}_0, \mathbf{E}_1, \mathbf{V}_1, \mathbf{E}_2\}$. Then by applying the matrix we obtain

$$\begin{bmatrix} \mathbf{E}_0 \\ \mathbf{V}_0 \\ \mathbf{E}_1 \\ \mathbf{V}_1 \\ \mathbf{E}_2 \end{bmatrix} = \frac{1}{8} \begin{bmatrix} 4 & 4 & 0 & 0 \\ 1 & 6 & 1 & 0 \\ 0 & 4 & 4 & 0 \\ 0 & 1 & 6 & 1 \\ 0 & 0 & 4 & 4 \end{bmatrix} \begin{bmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{bmatrix}$$

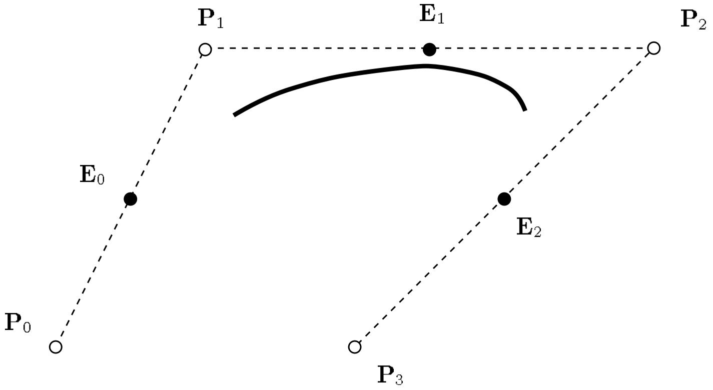
The edge points $\mathbf{E}_0$, $\mathbf{E}_1$ and $\mathbf{E}_2$ are calculated by

$$\begin{aligned} \mathbf{E}_0 &= \frac{1}{8}(4\mathbf{P}_0 + 4\mathbf{P}_1) \\ &= \frac{\mathbf{P}_0 + \mathbf{P}_1}{2} \end{aligned}$$

$$\begin{aligned} \mathbf{E}_1 &= \frac{1}{8}(4\mathbf{P}_1 + 4\mathbf{P}_2) \\ &= \frac{\mathbf{P}_1 + \mathbf{P}_2}{2} \end{aligned}$$

$$\begin{aligned} \mathbf{E}_2 &= \frac{1}{8}(4\mathbf{P}_2 + 4\mathbf{P}_3) \\ &= \frac{\mathbf{P}_2 + \mathbf{P}_3}{2} \end{aligned}$$

and indeed are the midpoints of the line segments connecting the original control points. These points are illustrated in the following figure:

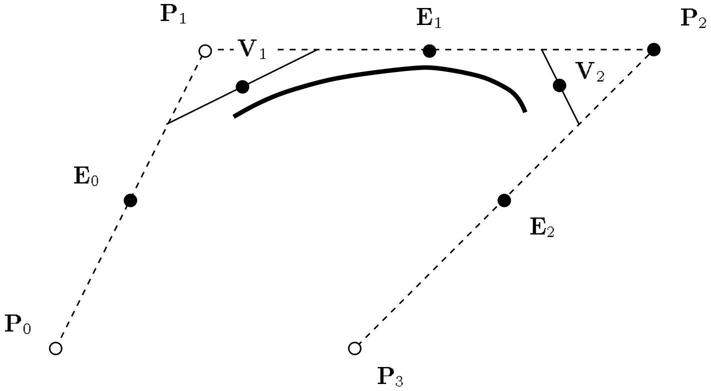


The vertex points $\mathbf{V}_0$ and $\mathbf{V}_1$ are calculated by

$$\mathbf{V}_0 = \frac{1}{8}(\mathbf{P}_0 + 6\mathbf{P}_1 + \mathbf{P}_2)$$

$$
\begin{aligned}
&= \frac{1}{8}((\mathbf{P}_0 + \mathbf{P}_1) + 4\mathbf{P}_1 + (\mathbf{P}_1 + \mathbf{P}_2)) \\
&= \frac{1}{4}(\frac{\mathbf{P}_0 + \mathbf{P}_1}{2} + 2\mathbf{P}_1 + \frac{\mathbf{P}_1 + \mathbf{P}_2}{2}) \\
&= \frac{1}{4}(\mathbf{E}_0 + 2\mathbf{P}_1 + \mathbf{E}_1) \\
&= \frac{\frac{\mathbf{E}_0+\mathbf{P}_1}{2} + \frac{\mathbf{P}_1+\mathbf{E}_1}{2}}{2}
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{V}_1 &= \frac{1}{8}(\mathbf{P}_1 + 6\mathbf{P}_2 + \mathbf{P}_3) \\
&= \frac{1}{8}((\mathbf{P}_1 + \mathbf{P}_2) + 4\mathbf{P}_2 + (\mathbf{P}_2 + \mathbf{P}_3)) \\
&= \frac{1}{4}(\frac{\mathbf{P}_1 + \mathbf{P}_2}{2} + 2\mathbf{P}_2 + \frac{\mathbf{P}_2 + \mathbf{P}_3}{2}) \\
&= \frac{1}{4}(\mathbf{E}_1 + 2\mathbf{P}_2 + \mathbf{E}_2) \\
&= \frac{\frac{\mathbf{E}_1+\mathbf{P}_2}{2} + \frac{\mathbf{P}_2+\mathbf{E}_2}{2}}{2}
\end{aligned}
$$



and each is the midpoint of the line segment that joins the respective midpoints of the line segments from the vertex point to the adjacent edge points. We note that the calculations require only midpoints of line segments to be determined.

2.2. The General Case of Arbitrary Length Control Polygons

Extending the above refinement rules to control polygons of arbitrary length is straightforward. Given a control polygon $\{\mathbf{P}_0, \mathbf{P}_1, ..., \mathbf{P}_n\}$, we extend the refinement generated for four points by defining the refined control polygon to be

$$\{\mathbf{E}_0, \mathbf{V}_0, \mathbf{E}_1, \mathbf{V}_1, \mathbf{E}_2, ..., \mathbf{E}_{n-2}, \mathbf{V}_{n-2}, \mathbf{E}_{n-1}\}$$

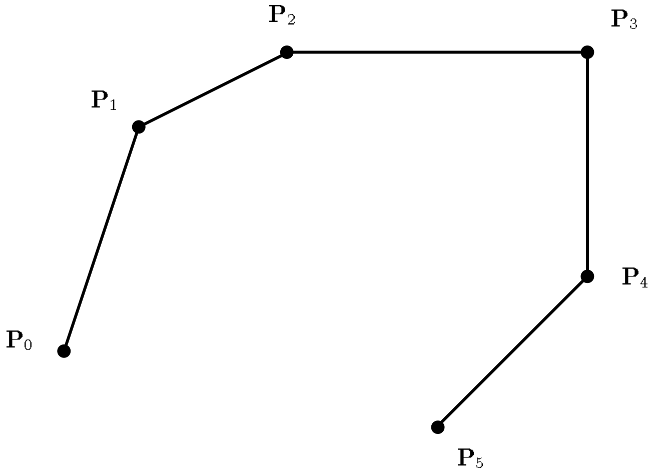
where each $\mathbf{E}_i$, an edge point, lies on the edge of the original control polygon, and each $\mathbf{V}_i$, a vertex point, corresponds to an internal vertex of the original control polygon. The new control polygon has $2n-1$ control points (the original has $n+1$). The rules to calculate the vertex and edge points are

$$\mathbf{E}_i = \frac{\mathbf{P}_i + \mathbf{P}_{i+1}}{2}$$

and

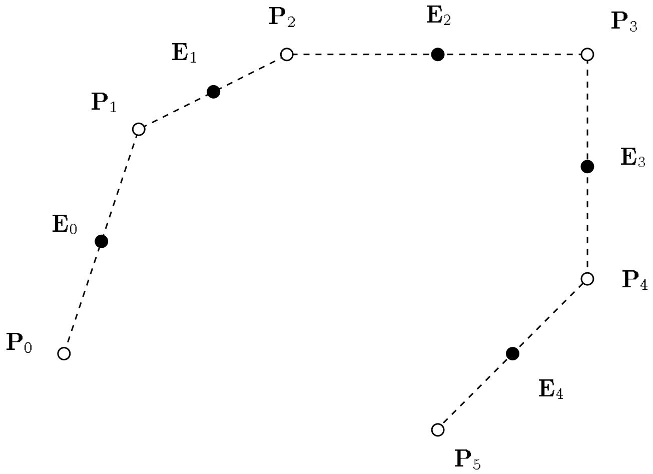
$$\mathbf{V}_i = \frac{\mathbf{E}_i + 2\mathbf{P}_i + \mathbf{E}_{i+1}}{4}$$

It is fairly easy to see these how these rules work by considering the following example: If we are given the control polygon below.



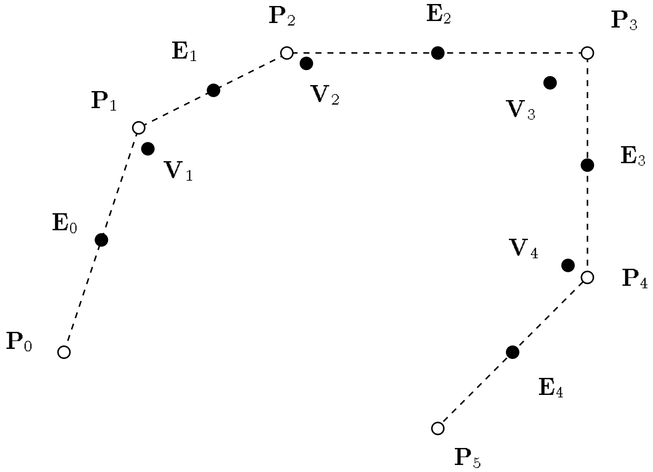
The edge points are calculated as the midpoints of the line segments forming the control polygon by

$$\mathbf{E}_i = \frac{\mathbf{P}_i + \mathbf{P}_{i+1}}{2}$$

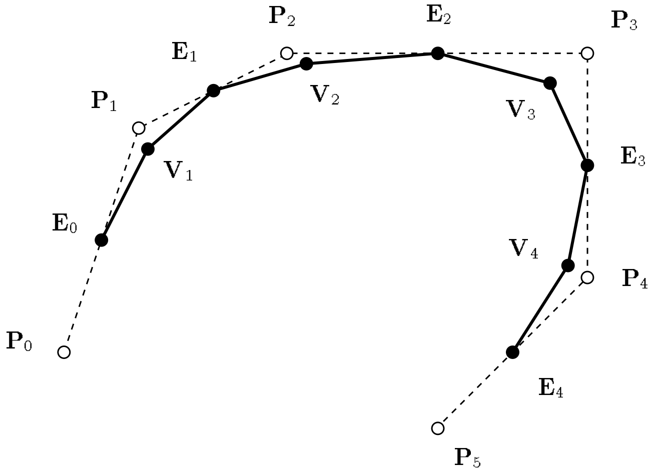


and the vertex points are calculated as the average of twice the control point and the two edge points adjacent to the control point by

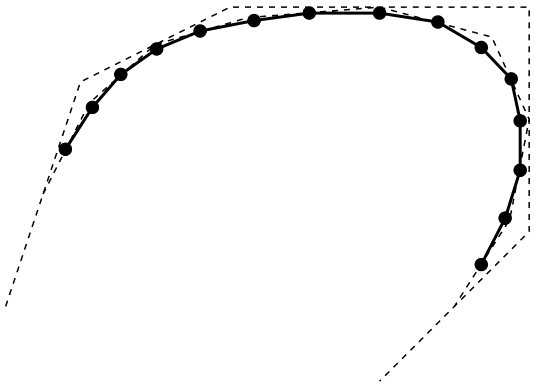
$$\mathbf{V}_i = \frac{\mathbf{E}_i + 2\mathbf{P}_i + \mathbf{E}_{i+1}}{4}$$



Connecting the edges and vertex points generated by the refinement gives the new control polygon.



This new set of edge points and vertex points can be considered a new control polygon, and by applying the refinement rules again, we generate the second refinement shown below.



One can see the curve actually taking shape now, and it is easy to see that if we continue this process, repetitively generating new control polygons by generating edge points and vertex points, then the successive control polygons will converge to the curve.

Thus the refinement rules for cubic uniform B-spline curves can be defined by utilizing the edge points and vertex points defined above. To differentiate from the processes we develop for the bivariate and trivariate cases, we will call the rule by which the edge points are calculated the *curve-edge rule* and the rule by which the vertex points are calculated the *curve-vertex rule.*

2.3. Summary of the Classification in the Univariate Case

We have shown that in the curve case, we can classify the points of the refined mesh into two types.

- edge points – those that can be calculated by the *curve-edge rule*: the points are the average of the two vertices that define the edge.
- vertex points – those that can be calculated by the *curve-vertex rule*: the points are an affine combination of the two edge points for the edges radiating from this vertex and the point itself with weights $\frac{1}{4}$, $\frac{1}{4}$, and $\frac{1}{2}$, respectively.

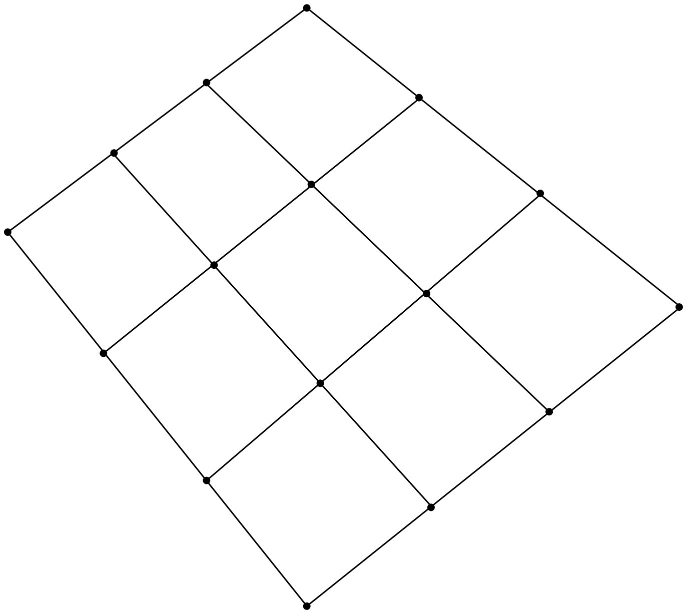
3. The Bivariate Case

Consider a bicubic uniform B-spline patch defined by a control point mesh

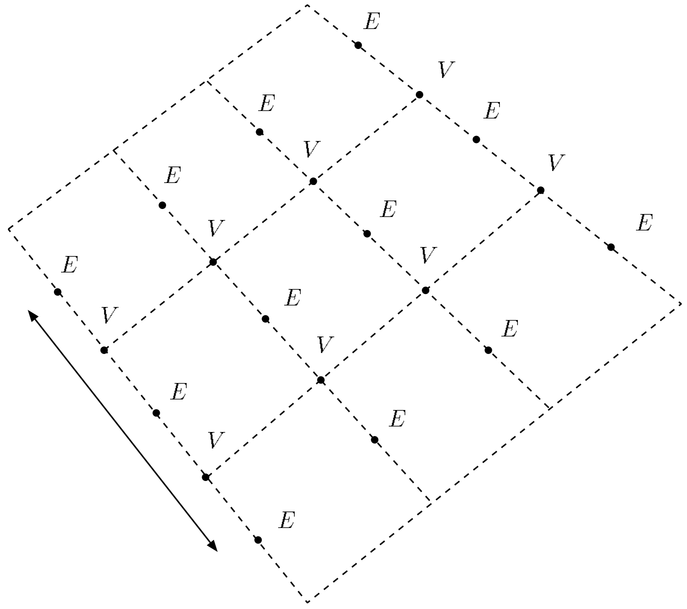
$$\{\mathbf{P}_{i,j} \ : \ 0 \leq i \leq n_1, 0 \leq j \leq n_2\}$$

This is a *tensor product surface*, which enables us to consider each respective parameter in turn to generate the refinement – i.e. utilize the univariate refinement to generate a refinement of all rows of the control point mesh; take the resulting points and apply the univariate refinement to the columns – utilizing the univariate refinement rules to define the points of the bivariate refinement.

Consider the control point mesh given below,



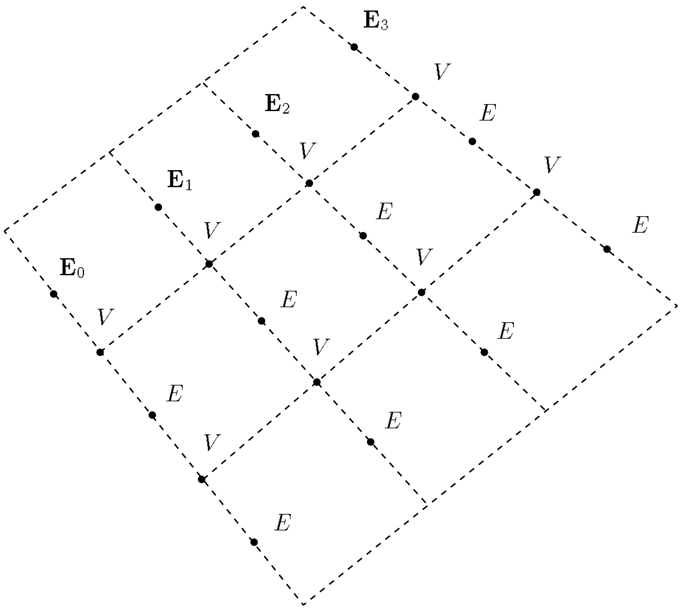
which we have kept simple for purposes of illustration. If we utilize the univariate refinement rules to refine the rows of the mesh, we obtain the following points (where the new points all have been labeled as to the rule that was used in generating them – curve-edge(E) or curve-vertex(V)).



Taking now the new mesh consisting of the points labeled E or V (which is now 5×4), we will refine the columns of this mesh to obtain the refinement in the bivariate case. Examining this closely, we note that two cases arise : a column contains either four points that were generated using the curve-edge rule on the rows, or four points that were generated by the curve-vertex rule. We consider each case separately.

3.1. Case I – All Points of a Column were Generated by the Curve-Edge Rule on the Rows of the Mesh

Consider the case where the control points in the column were all generated by the curve-edge rule from the univariate case. For example, this could be from the highlighted row of points below:



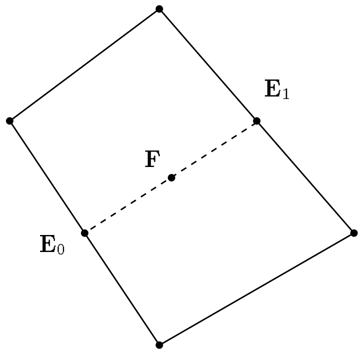
These points will be refined according to the univariate rules – i.e. either the curve-edge rule, or the curve-vertex rule.

3.1.1. Using the Curve-Edge Rule

Consider points $\mathbf{E}_0$ and $\mathbf{E}_1$. A new point of the refined mesh is calculated by using the curve-edge rule – defining a new point as the midpoint of $\mathbf{E}_0$ and $\mathbf{E}_1$. That is,

$$\mathbf{F} = \frac{\mathbf{E}_0 + \mathbf{E}_1}{2}$$

The following figure, which represents the mesh in the area of the points, illustrates this calculation. Since the edge points $\mathbf{E}_0$ and $\mathbf{E}_1$ are the midpoints of their respective edges and $\mathbf{F}$ is the midpoint of $\mathbf{E}_0$ and $\mathbf{E}_1$, it is easy to see that $\mathbf{F}$ is just the average of all the mesh points that surround the face in which $\mathbf{F}$ lies.

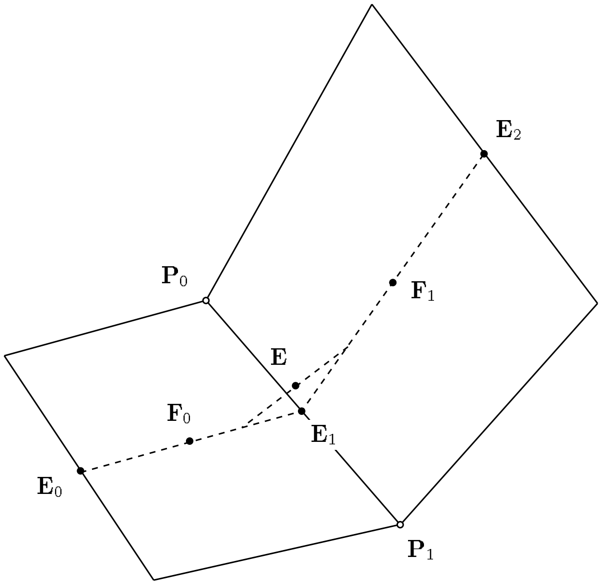


This point will be called a “face point”. It is calculated by the *patch-face rule*, which is to take the average of all control points that surround the face in which this point lies. These face points are calculated for each pair of control points in the column.

3.1.2. Using the Curve-Vertex Rule

Consider points $\mathbf{E}_0$, $\mathbf{E}_1$ and $\mathbf{E}_2$. A new point of the refined mesh is calculated by using the curve-vertex rule – that is

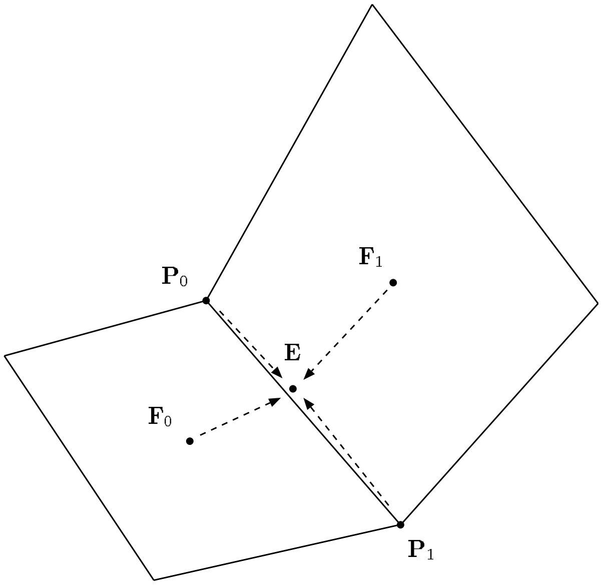
$$\mathbf{E} = \frac{\frac{\mathbf{E}_0+\mathbf{E}_1}{2} + 2\mathbf{E}_1 + \frac{\mathbf{E}_1+\mathbf{E}_2}{2}}{2}$$



From the discussion in the previous section (3.1.1), the two midpoints in the numerator of the above equation are just face points, and can be calculated by the patch-face rule. This implies that

$$\begin{aligned}\mathbf{E} &= \frac{\frac{\mathbf{E}_0+\mathbf{E}_1}{2} + 2\mathbf{E}_1 + \frac{\mathbf{E}_1+\mathbf{E}_2}{2}}{2} \\ &= \frac{\mathbf{F}_0 + \mathbf{F}_1 + 2\mathbf{E}_1}{4} \\ &= \frac{\mathbf{F}_0 + \mathbf{F}_1 + \mathbf{P}_0 + \mathbf{P}_1}{4}\end{aligned}$$

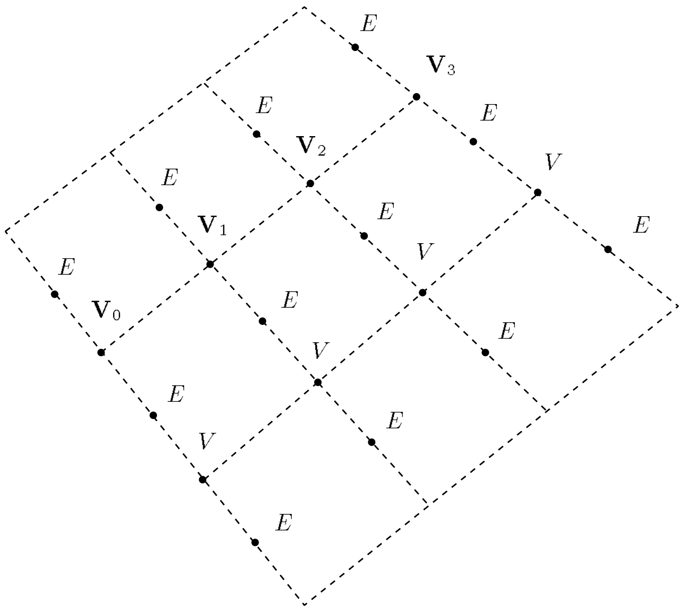
where $\mathbf{F}_0$ and $\mathbf{F}_1$ are the face points for the two faces adjacent to the edge, and $\mathbf{P}_0$ and $\mathbf{P}_1$ are the endpoints of the edge. This is shown in the following figure.



This new point is commonly called an "edge point" (It is associated with the edge $\overline{\mathbf{P}_0\mathbf{P}_1}$. It is calculated by the *patch-edge rule*, which states that the new point is the average of four points: the two endpoints of the edge and the two face points of the faces adjacent to the edge. These edge points are calculated for each triple of points $\mathbf{E}_0, \mathbf{E}_1, \mathbf{E}_2$ and $\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3$ in the row.

3.2. Case II – All Points Generated by the Curve-Vertex Rule on the Rows of the Mesh

Consider the case where the control points of the column were all generated by the curve-vertex rule from the univariate case. For example, this could be from the highlighted row of points in the following illustration:



The new points are calculated according to the univariate rules, and there are two cases to consider.

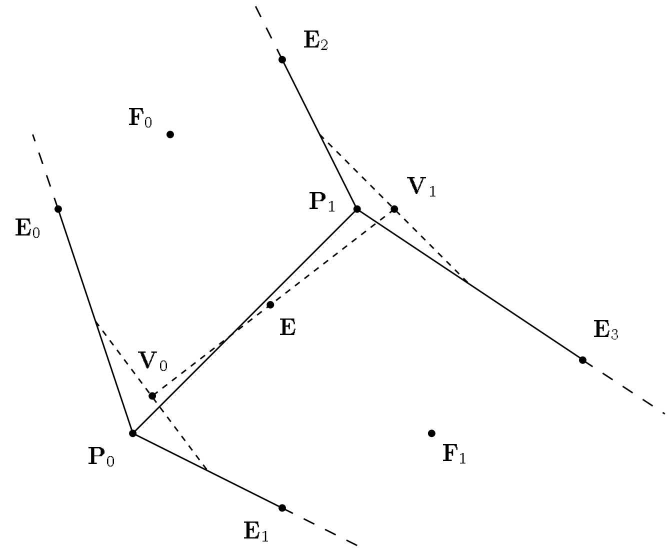
3.2.1. Using the Curve-Edge Rule

Consider points $\mathbf{V}_0$ and $\mathbf{V}_1$. A new point of the refined mesh is calculated by using the curve-edge rule from the univariate case – that is,

$$\mathbf{E} \;=\; \frac{\mathbf{V}_0 + \mathbf{V}_1}{2}$$

The following figure illustrates the local area around such a point and the calculation procedure.[1]

[1] Note that we have labeled the points of the mesh surrounding $\mathbf{V}_0$ and $\mathbf{V}_1$ so that they represent how they were calculated – $\mathbf{F}$ for face points, etc.



Note that each vertex point is calculated from edge points and original vertices of the mesh as follows:

$$\begin{aligned}\mathbf{V}_0 &= \frac{\mathbf{E}_0 + \mathbf{E}_1 + 2\mathbf{P}_0}{4} \\ \mathbf{V}_1 &= \frac{\mathbf{E}_2 + \mathbf{E}_3 + 2\mathbf{P}_1}{4}\end{aligned}$$

where the points are labeled as in the illustration above. This implies that

$$\begin{aligned}\mathbf{E} &= \frac{\mathbf{V}_0 + \mathbf{V}_1}{2} \\ &= \frac{\frac{\mathbf{E}_0+\mathbf{E}_1+2\mathbf{P}_0}{4} + \frac{\mathbf{E}_2+\mathbf{E}_3+2\mathbf{P}_1}{4}}{2} \\ &= \frac{\frac{\mathbf{E}_0+\mathbf{E}_2}{2} + \frac{\mathbf{E}_1+\mathbf{E}_3}{2} + \mathbf{P}_0 + \mathbf{P}_1}{4} \\ &= \frac{\mathbf{F}_0 + \mathbf{F}_1 + \mathbf{P}_0 + \mathbf{P}_1}{4}\end{aligned}$$

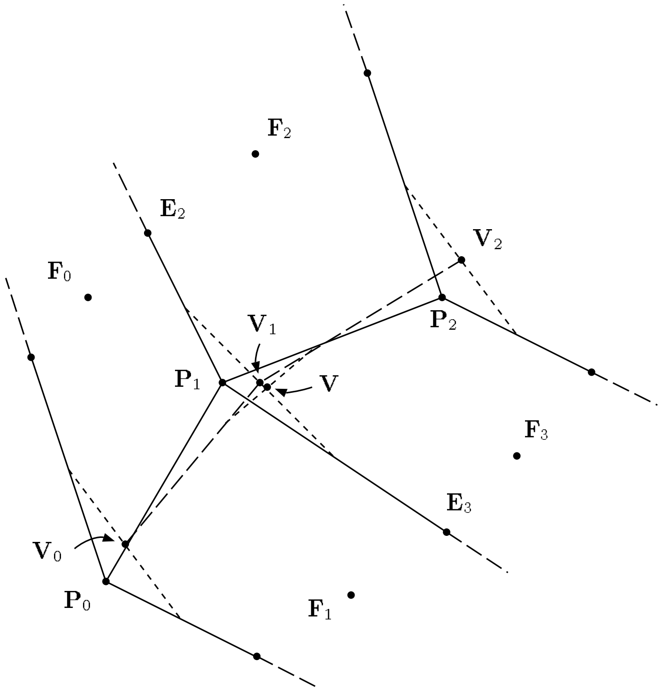
and therefore this point is an “edge point” and can be calculated by the patch-edge rule developed in the previous section (3.1.2) – i.e. calculating the average of the two adjacent face points along with the two original control points of the mesh that are the endpoints of the edge.

3.2.2. Using the Curve-Vertex Rule

Consider points $\mathbf{V}_0$, $\mathbf{V}_1$ and $\mathbf{V}_2$. A new point of the refined mesh is calculated by using the curve-vertex rule of the univariate case – that is,

$$\mathbf{V} = \frac{\frac{\mathbf{V}_0+\mathbf{V}_1}{2} + 2\mathbf{V}_1 + \frac{\mathbf{V}_1+\mathbf{V}_2}{2}}{4}$$

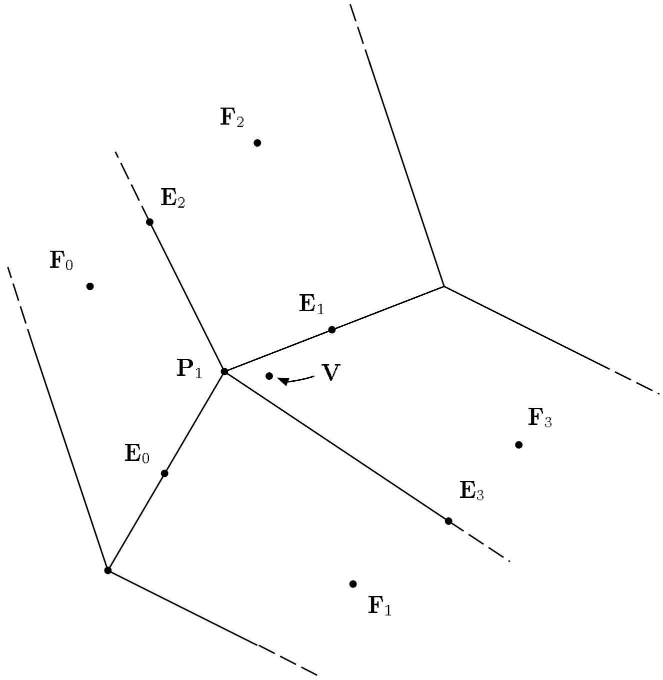
The following figure illustrates the local area about such a point and the calculation procedure.



The points $\frac{\mathbf{V}_0+\mathbf{V}_1}{2}$ and $\frac{\mathbf{V}_1+\mathbf{V}_2}{2}$, as is in section 3.1.2, can be calculated by the patch-edge rule, and so

$$\begin{aligned}
\mathbf{V} &= \frac{\frac{\mathbf{V}_0+\mathbf{V}_1}{2} + 2\mathbf{V}_1 + \frac{\mathbf{V}_1+\mathbf{V}_2}{2}}{4} \\
&= \frac{\frac{\mathbf{F}_0+\mathbf{F}_1+\mathbf{P}_0+\mathbf{P}_1}{4} + \frac{\mathbf{F}_2+\mathbf{F}_3+\mathbf{P}_1+\mathbf{P}_2}{4} + 2\mathbf{V}_1}{4} \\
&= \frac{\frac{\mathbf{F}_0+\mathbf{F}_1+\mathbf{F}_2+\mathbf{F}_3}{4} + \frac{\mathbf{P}_0+2\mathbf{P}_1+\mathbf{P}_2}{4} + 2\left(\frac{\mathbf{E}_2+\mathbf{E}_3+2\mathbf{P}_1}{4}\right)}{4} \\
&= \frac{\frac{\mathbf{F}_0+\mathbf{F}_1+\mathbf{F}_2+\mathbf{F}_3}{4} + 2\left(\frac{\mathbf{E}_0+\mathbf{E}_1+\mathbf{E}_2+\mathbf{E}_3}{4}\right) + \mathbf{P}_1}{4}
\end{aligned}$$

where the points $\mathbf{E}_0$, $\mathbf{E}_1$, $\mathbf{E}_2$ and $\mathbf{E}_3$ are midpoints of the line segments that radiate from $\mathbf{P}_1$, and are shown in the following figure.



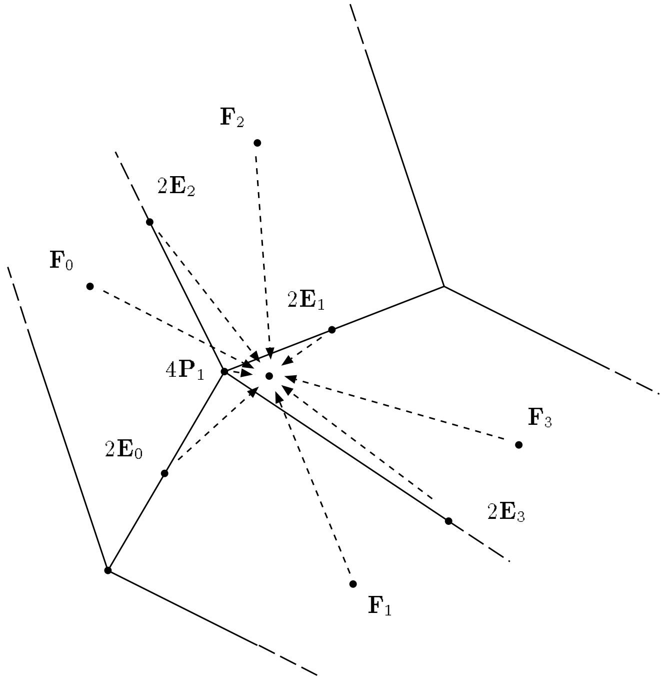
This has generated the *patch-vertex rule* which states that the vertex point is an affine combination of the values

- the average of the face points for the faces that are adjacent to the vertex,
- the average of the midpoints of the edges that radiate from the vertex,
- the original vertex in the control point mesh.

with weights of $\frac{1}{4}$, $\frac{1}{2}$ and $\frac{1}{4}$, respectively. By multiplying numerator and denominator by 4, we obtain

$$\mathbf{V} \;=\; \frac{\mathbf{F}_0+\mathbf{F}_1+\mathbf{F}_2+\mathbf{F}_3+2\mathbf{E}_0+2\mathbf{E}_1+2\mathbf{E}_2+2\mathbf{E}_3+4\mathbf{P}_1}{16}$$

and so the influence of the points on the vertex point is shown as follows:



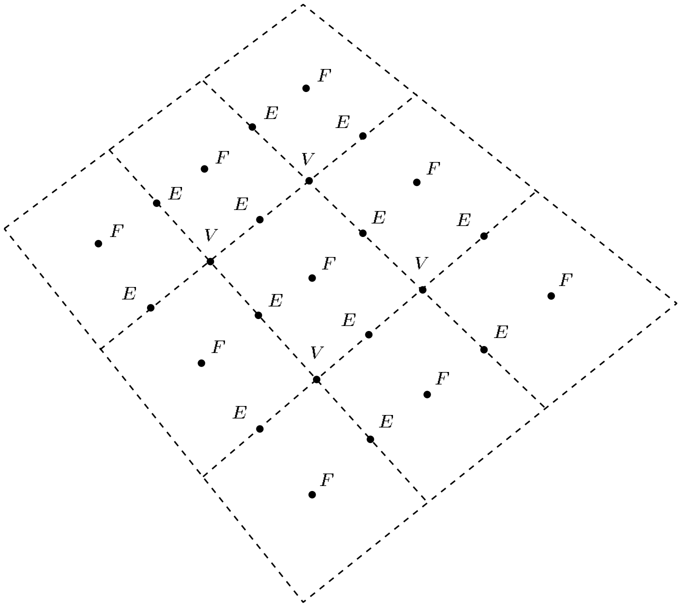
One each of these vertex points must be calculated for each triple $\mathbf{V}_0, \mathbf{V}_1, \mathbf{V}_2$ and $\mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3$.

3.3. Summary of the Classification in the Bivariate Case

We have shown that in the bivariate case, the points of a refinement can be classified into three types.

- face points – those that can be calculated by the *patch-face rule* : the points are the average of the control points in the mesh that surround a face.
- edge points – those that can be calculated by the *patch-edge rule* : the points are the average of the two face points for the faces containing the edge and the two control points that are the endpoints of the edge.
- vertex points – those that can be calculated by the *patch-vertex rule* : the points an affine combinations of the face point average, the edge point average and the original control point that corresponds to this vertex, with weights of $\frac{1}{4}$, $\frac{1}{2}$, and $\frac{1}{4}$, respectively.

These points are illustrated in our simple example below. Each point has been labeled corresponding to its calculation method (V, E, or F).



4. Catmull-Clark Surfaces

Ed Catmull and Jim Clark [1] were the first to notice that the classification of refinement points as “face”, “vertex”, or “edge” points could be generalized to work with meshes of arbitrary topology. They defined the following procedure

- For each face of the mesh, generate a new face point – which is the average of all the control points defining the face (Note that faces may have 3, 4, 5, or many points now defining them).
- For each edge of the mesh, generate a new edge point – which is calculated as the average of the endpoints of the edge with the two new face points of the faces adjacent to the edge.
- For each internal vertex of the mesh, calculate a new vertex point – which is calculated as the average

  $$\mathbf{V} = \frac{\mathbf{Q} + 2\mathbf{R} + (n-3)\mathbf{P}}{n}$$

  where $\mathbf{Q}$ is the average of the face points of all faces adjacent to the vertex, $\mathbf{R}$ is the average of the midpoints of all edges incident on the vertex, $\mathbf{P}$ is the vertex itself, and n is the number of edges radiating from the vertex.

  Note that this rule is the same as the rule for the uniform cubic B-spline case when $n = 4$. It has been extended by realizing that the average at a vertex should be based upon the number of edges that radiate from it.

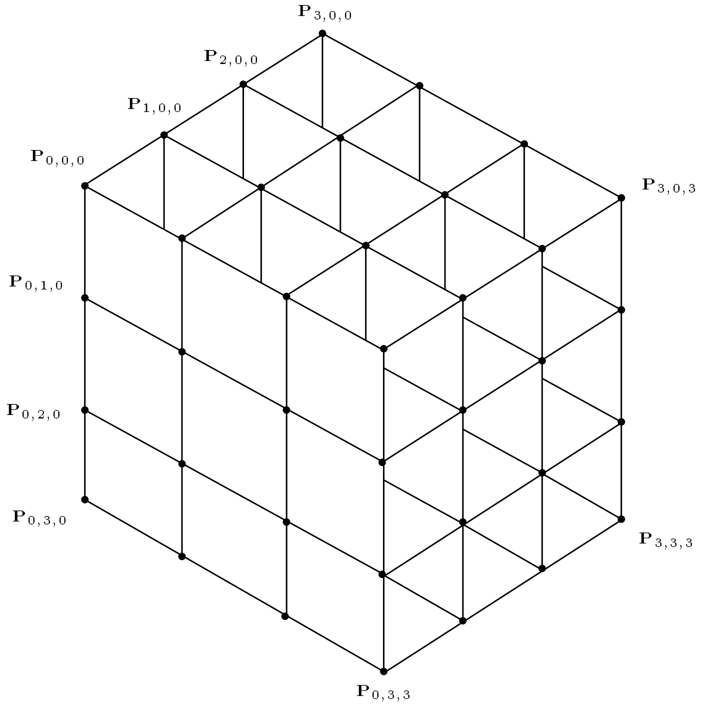
To reconnect the mesh after these rules have been applied, we first connect each new face point to the new edge points of the edges defining the original face, and then connect each new vertex point to the new edge points of all edges incident on the original control point.

5. The Trivariate Case

Consider a hexahedral lattice

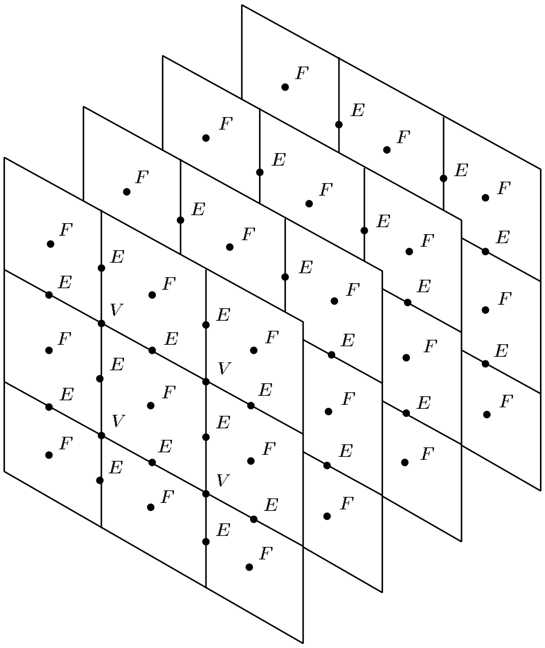
$$\{\mathbf{P}_{i,j,k} \,:\, 0 \le i \le n_1, 0 \le j \le n_2, 0 \le k \le n_3\}$$

as is shown (in simplified form) below.



This lattice can be used to form a trivariate cubic uniform B-spline solid. Since this is a tensor-product solid, we can generate refinement rules for the solid by utilizing the bivariate refinement rules on each "plane" of the lattice, coupled with the univariate rules in the other dimension.

Consider the refined control lattice below where we have refined each mesh according to the bivariate rules on each plane of the figure – generating face points (F), edge points (E) and vertex points (V). We will use these points and the rules for the univariate case to generate the refinement according to the third parameter.



If we wish to refine according to the other parameter, we end up with three cases : the four control points to be used are all face points of the bivariate refinement; the four control points to be used are all edge points of the bivariate refinement; or the four control points to be used are all vertex points of the bivariate refinement. We treat each case separately.

5.1. Case I – All Face Points

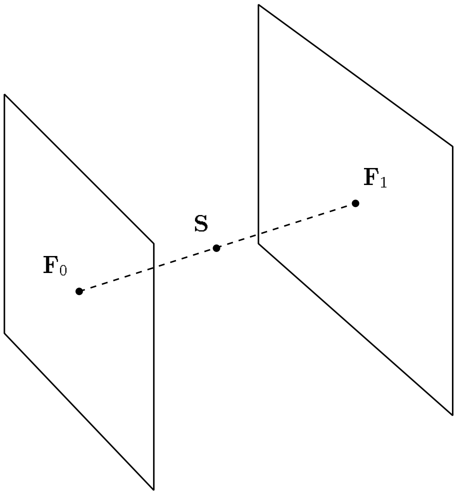
Consider the case where univariate refinement must be done on a list of control points all of which are face points generated by the bivariate refinement. From the univariate case, these can be refined according to two different rules: the curve-edge rule and the curve-vertex rule.

5.1.1. Using the Curve-Edge Rule

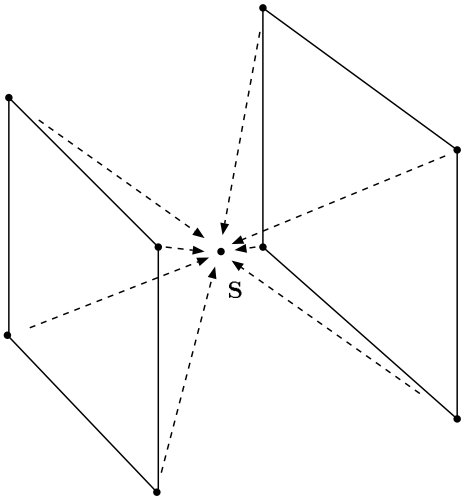
Consider the control points $\mathbf{F}_0$ and $\mathbf{F}_1$. A new point of the refined mesh is calculated by the univariate curve-edge rule – that is the new point is the midpoint of the line segment defined by $\mathbf{F}_0$ and $\mathbf{F}_1$

$$\mathrm{S} \;=\; \frac{\mathbf{F}_0 + \mathbf{F}_1}{2}$$

The following figure illustrates the local mesh around this point and the calculation procedure



Since the face points are average of the points surrounding a face, this point is then the average of all points surrounding the hexahedral solid containing the point. This point will be called a "solid point". It is calculated by the *solid-solid rule*, which is to take the average of all control points that surround the hexahedral solid in which this point lies.



5.1.2. Using the Curve-Vertex Rule

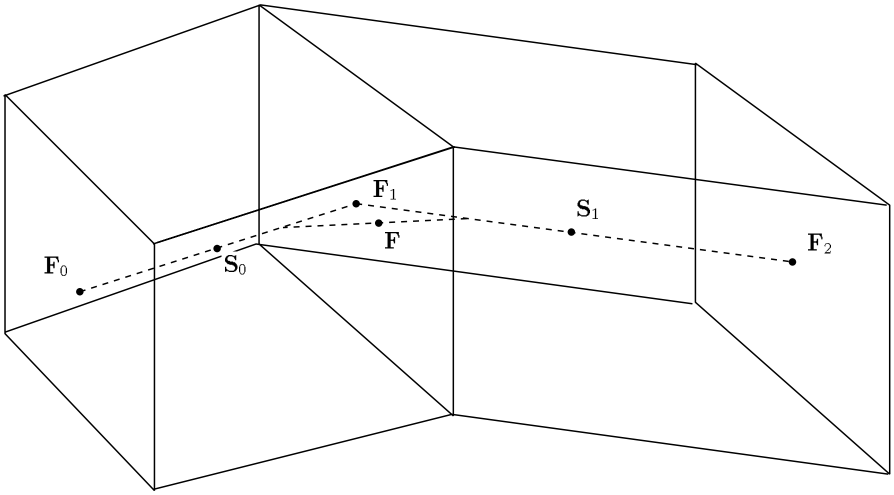
Consider the control points $\mathbf{F}_0$, $\mathbf{F}_1$ and $\mathbf{F}_2$. A new point of the refinement is calculated by the univariate curve-vertex rule – that is

$$\mathbf{F} = \frac{\frac{\mathbf{F}_0+\mathbf{F}_1}{2} + 2\mathbf{F}_1 + \frac{\mathbf{F}_1+\mathbf{F}_2}{2}}{4}$$

By the above calculations (section 5.1.1), the two midpoints in the numerator are both "solid points" and therefore this point can be calculated as

$$\mathbf{F} = \frac{\mathbf{S}_0 + 2\mathbf{F}_1 + \mathbf{S}_1}{4}$$

as is shown in the illustration below:



These points are called "face" points, as they are associated with a face of the control point mesh. They are calculated by the *solid-face* rule which states that they are an affine combinations of three points – the two solid points of the hexahedra containing the face and the face point calculated by the patch-face rule – with weights of $\frac{1}{4}$, $\frac{1}{4}$ and $\frac{1}{2}$, respectively

5.2. Case II – All Edge Points

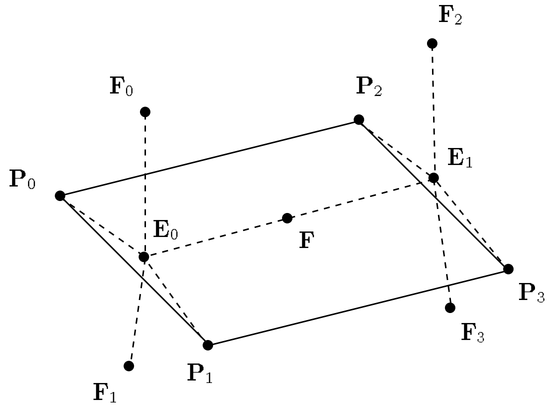
Consider the case where univariate refinement must be done on a list of control points all of which are edge points of the bivariate refinement. From the univariate case, these can be refined according to two different rules: the curve-edge rule and the curve-vertex rule.

5.2.1. Using the Curve-Edge Rule

Consider the control points $\mathbf{E}_0$ and $\mathbf{E}_1$. A new point of the refined mesh is calculated by using the univariate curve-edge rule. That is, the new point is the midpoint of the line segment defined by $\mathbf{E}_0$ and $\mathbf{E}_1$.

$$\mathbf{F} = \frac{\mathbf{E}_0 + \mathbf{E}_1}{2}$$

The following figure illustrates the local portion of the control mesh about this point and the calculation procedure.



Since each of $\mathbf{E}_0$ and $\mathbf{E}_1$ are edge points in the bivariate scheme, they are calculated according to the patch-edge rule and are the average of the endpoints of the edge and the face points of the faces that are adjacent to the edge. Substituting this into the equation above, we obtain

$$\begin{aligned}\mathbf{F} &= \frac{\frac{\mathbf{F}_0+\mathbf{F}_1+\mathbf{P}_0+\mathbf{P}_1}{4} + \frac{\mathbf{F}_2+\mathbf{F}_3+\mathbf{P}_2+\mathbf{P}_3}{4}}{2} \\ &= \frac{\frac{\mathbf{F}_0+\mathbf{F}_2}{2} + \frac{\mathbf{F}_1+\mathbf{F}_3}{2} + 2\left(\frac{\mathbf{P}_0+\mathbf{P}_1+\mathbf{P}_2+\mathbf{P}_3}{4}\right)}{4}\end{aligned}$$

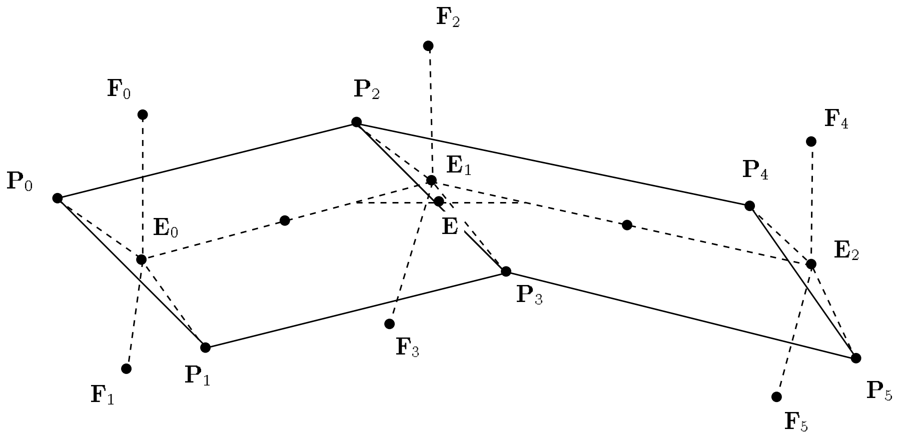
which is the average of the two solid points that are adjacent to the face and twice the face point – which is just the solid-face rule defined in section 5.1.2.

5.2.2. Using the Curve-Vertex Rule

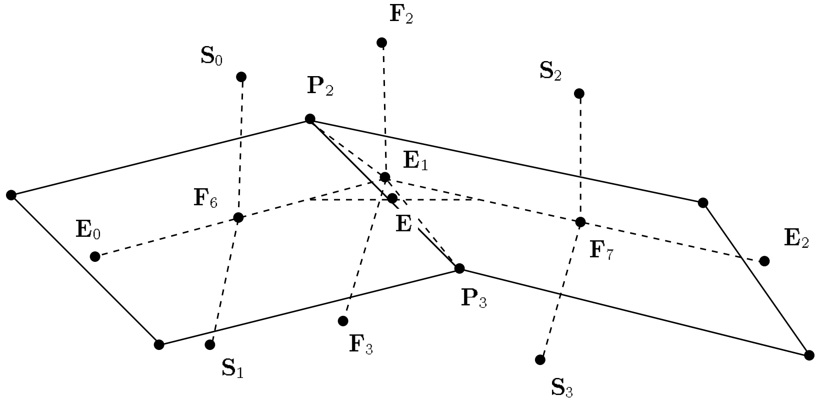
Consider the control points $\mathbf{E}_0$, $\mathbf{E}_1$ and $\mathbf{E}_2$. A new point of the refined mesh by is calculated using the univariate curve-vertex rule. That is

$$\mathbf{E} = \frac{\frac{\mathbf{E}_0+\mathbf{E}_1}{2} + 2\mathbf{E}_1 + \frac{\mathbf{E}_1+\mathbf{E}_2}{2}}{4}$$

The local portion of the control mesh about such a point and the calculation procedure are shown in the illustration below:



Since each of $\mathbf{E}_0$, $\mathbf{E}_1$ and $\mathbf{E}_2$ are edge points in the bivariate scheme, they are calculated according to the patch-edge rule and are the average of the endpoints of the edge and the face points of the faces that are adjacent to the edge. The two fractions in the numerator are both face points in the trivariate scheme and can be calculated by the solid-face rule (see section 5.1.2). Using the notation in the figure below, and substituting, gives



$$\begin{aligned}\mathbf{E} &= \frac{\frac{\mathbf{S}_0+\mathbf{S}_1+2\mathbf{F}_6}{4} + 2\left(\frac{\mathbf{F}_2+\mathbf{F}_3+\mathbf{P}_2+\mathbf{P}_3}{4}\right) + \frac{\mathbf{S}_2+\mathbf{S}_3+2\mathbf{F}_7}{4}}{4} \\ &= \frac{\frac{\mathbf{S}_0+\mathbf{S}_1+\mathbf{S}_2+\mathbf{S}_3}{4} + 2\left(\frac{\mathbf{F}_2+\mathbf{F}_3+\mathbf{F}_6+\mathbf{F}_7}{4}\right) + \frac{\mathbf{P}_2+\mathbf{P}_3}{2}}{4}\end{aligned}$$

That is, this point can be calculated by taking the average of the following values

- the average of the solid points for the hexahedra that contain this edge.
- twice the average of the face points for those faces that contain this edge.

- The midpoint of the edge.

This point is associated with an edge of the original mesh and thus will be called an “edge point”. The calculation of this point will be called the *solid-edge rule.*

5.3. Case III – All Vertex Points

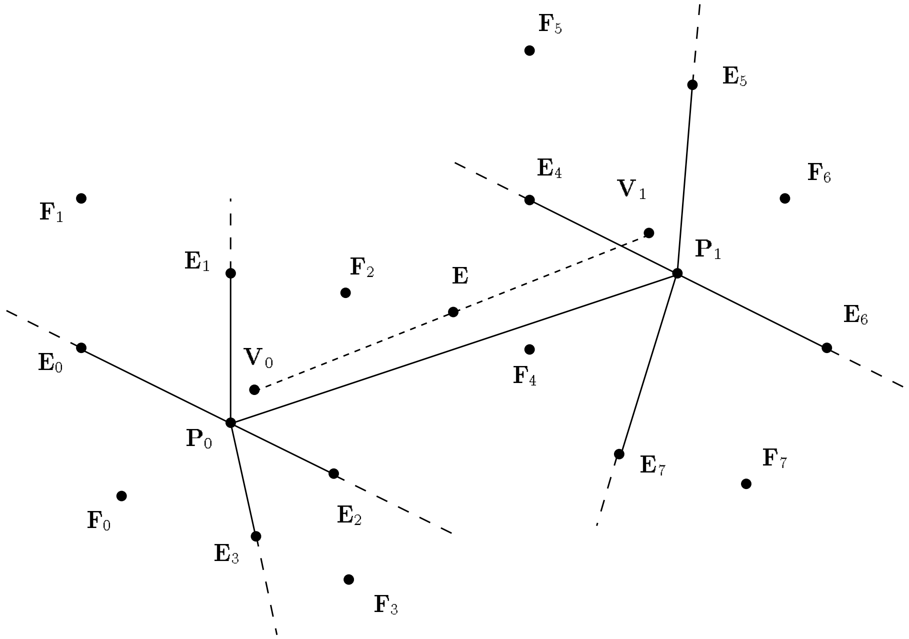
Consider the case where univariate refinement must be done on a list of control points, all of which are vertex points of the bivariate refinement. From the univariate case, these can be refined according to two different rules: the curve-edge rule and the curve-vertex rule.

5.3.1. Using the Curve-Edge Rule

Consider the control points $\mathbf{V}_0$ and $\mathbf{V}_1$. A new point of the refined mesh is calculated by using the univariate curve-edge rule – the new point is the midpoint of the line segment defined by the points $\mathbf{V}_0$ and $\mathbf{V}_1$. That is,

$$\mathbf{E} = \frac{\mathbf{V}_0 + \mathbf{V}_1}{2}$$

The following figure illustrates the local portion of the mesh about such a point and the calculation procedure.



Since each of $\mathbf{V}_0$ and $\mathbf{V}_1$ are vertex points in the bivariate scheme, they are calculated according to the patch-vertex rule and can be written as

$$\begin{aligned} \mathbf{V}_0 &= \frac{\frac{\mathbf{F}_0+\mathbf{F}_1+\mathbf{F}_2+\mathbf{F}_3}{4} + 2\left(\frac{\mathbf{E}_0+\mathbf{E}_1+\mathbf{E}_2+\mathbf{E}_3}{4}\right) + \mathbf{P}_0}{4} \\ \mathbf{V}_1 &= \frac{\frac{\mathbf{F}_4+\mathbf{F}_5+\mathbf{F}_6+\mathbf{F}_7}{4} + 2\left(\frac{\mathbf{E}_4+\mathbf{E}_5+\mathbf{E}_6+\mathbf{E}_7}{4}\right) + \mathbf{P}_1}{4} \end{aligned}$$

where $\mathbf{F}_0$, $\mathbf{F}_1$, $\mathbf{F}_2$ and $\mathbf{F}_3$ are the patch face points of the respective faces that have $\mathbf{P}_0$ as a vertex, $\mathbf{F}_4$, $\mathbf{F}_5$, $\mathbf{F}_6$ and $\mathbf{F}_7$ are the patch face points of the respective faces that have $\mathbf{P}_1$ as a vertex, $\mathbf{E}_0$, $\mathbf{E}_1$, $\mathbf{E}_2$ and $\mathbf{E}_3$ are the patch edge points of the edges that radiate from $\mathbf{P}_0$, and $\mathbf{E}_4$, $\mathbf{E}_5$, $\mathbf{E}_6$ and $\mathbf{E}_7$ are the patch edge points of the edges that radiate from $\mathbf{P}_1$. Taking the average of these two quantities, one obtains

$$\begin{aligned} \mathbf{E} &= \frac{\mathbf{V}_0 + \mathbf{V}_1}{2} \\ &= \frac{\frac{\frac{\mathbf{F}_0+\mathbf{F}_4}{2}+\frac{\mathbf{F}_1+\mathbf{F}_5}{2}+\frac{\mathbf{F}_2+\mathbf{F}_6}{2}+\frac{\mathbf{F}_3+\mathbf{F}_7}{2}}{4} + 2\left(\frac{\frac{\mathbf{E}_0+\mathbf{E}_4}{2}+\frac{\mathbf{E}_1+\mathbf{E}_5}{2}+\frac{\mathbf{E}_2+\mathbf{E}_6}{2}+\frac{\mathbf{E}_3+\mathbf{E}_7}{2}}{4}\right) + \frac{\mathbf{P}_0+\mathbf{P}_1}{2}}{4} \end{aligned}$$

The first four midpoints in the numerator are all solid points calculated by the solid-solid rule (section 5.1.1), the second four midpoints in the numerator are all face points calculated by the patch-face rule (section 3.1.1), and the last midpoint is the edge point calculated by the curve-edge rule. Therefore, we have that this is just the solid-edge rule as defined in section 5.2.2.

5.3.2. Using the Curve-Vertex Rule

Consider the control points $\mathbf{V}_0$, $\mathbf{V}_1$ and $\mathbf{V}_2$. A new point of the refined mesh is calculated by using the univariate curve-vertex rule – i.e.

$$\mathbf{V} = \frac{\frac{\mathbf{V}_0+\mathbf{V}_1}{2} + 2\mathbf{V}_1 + \frac{\mathbf{V}_1+\mathbf{V}_2}{2}}{4}$$

By the calculations in section 5.2.2, we know that the first and the last fractions in the numerator are edge points and are calculated by the solid-edge rule. We also know that the vertex points are all calculated by the patch-vertex rule (section 3.2.2). Therefore the equation can be written as

$$\begin{aligned} \mathbf{V} &= \frac{\frac{\mathbf{V}_0+\mathbf{V}_1}{2} + 2\mathbf{V}_1 + \frac{\mathbf{V}_1+\mathbf{V}_2}{2}}{4} \\ &= \frac{1}{4}\left(\frac{\frac{\mathbf{S}_0+\mathbf{S}_1+\mathbf{S}_2+\mathbf{S}_3}{4} + 2\frac{\mathbf{F}_1+\mathbf{F}_2+\mathbf{F}_3+\mathbf{F}_4}{4} + \mathbf{E}_0}{4}\right. \\ &\qquad + 2\,\frac{\frac{\mathbf{F}_8+\mathbf{F}_9+\mathbf{F}_{10}+\mathbf{F}_{11}}{4} + 2\frac{\mathbf{E}_2+\mathbf{E}_3+\mathbf{E}_4+\mathbf{E}_5}{4} + \mathbf{P}_1}{4} \end{aligned}$$

$$
\begin{aligned}
& + \frac{\frac{\mathbf{S}_4+\mathbf{S}_5+\mathbf{S}_6+\mathbf{S}_7}{4} + 2\frac{\mathbf{F}_4+\mathbf{F}_5+\mathbf{F}_6+\mathbf{F}_7}{4} + \mathbf{E}_1}{4} \Bigg) \\
= \; & \frac{1}{16}\Bigg(2\left[\frac{\mathbf{S}_0 + \mathbf{S}_1 + \mathbf{S}_2 + \mathbf{S}_3 + \mathbf{S}_4 + \mathbf{S}_5 + \mathbf{S}_6 + \mathbf{S}_7}{8}\right] \\
& + 6\left[\frac{\mathbf{F}_0 + \mathbf{F}_1 + \mathbf{F}_2 + \mathbf{F}_3 + \mathbf{F}_4 + \mathbf{F}_5 + \mathbf{F}_6 + \mathbf{F}_7 + \mathbf{F}_8 + \mathbf{F}_9 + \mathbf{F}_{10} + \mathbf{F}_{11}}{12}\right] \\
& + 6\left[\frac{\mathbf{E}_0 + \mathbf{E}_1 + \mathbf{E}_2 + \mathbf{E}_3 + \mathbf{E}_4 + \mathbf{E}_5}{6}\right] \\
& + 2\,\mathbf{P}_1\Big)
\end{aligned}
$$

This is an affine combination of the four points

- the average of the eight solid points for each of the hexahedra that has $\mathbf{V}_1$ as a vertex,
- the average of the twelve face points for the faces that has $\mathbf{V}_1$ as a vertex,
- the average of the six edge points for the edges that radiate from $\mathbf{V}_1$, and
- $\mathbf{V}_1$ itself.

with weights $\frac{1}{8}$, $\frac{3}{8}$, $\frac{3}{8}$ and $\frac{1}{8}$ respectively. We will call this rule the *solid-vertex rule.*

5.4. Summary of the Classification in the Trivariate Case

We have shown that in the solid case, we can classify the refinement points into four types.

- solid points – those that can be calculated by the *solid-solid rule* : the points are the average of the control points in the lattice that bound the hexahedral volume surrounding this point.
- face points – those that can be calculated by the *solid-face rule* : the points can be written as

  $$\frac{\mathrm{S}_1 + \mathrm{S}_2 + 2\mathbf{F}}{4}$$

  where S_1 and S_2 are the solid points of the two hexahedra that contain the face and $\mathbf{F}$ is the face point calculated by the patch-face rule.
- edge points – those that can be calculated by the *solid-edge rule* : the points can be written as

  $$\frac{\mathrm{S} + 2\mathbf{F} + \mathbf{E}}{4}$$

  where S is the average of the solid points for those hexahedra that contain the edge, $\mathbf{F}$ is the average of the face points (patch-face rule) for those faces that contain the edge, and $\mathbf{E}$ is the midpoint of the edge.

- vertex points – those that can be calculated by the solid-vertex rule : the points can be written as

$$\frac{\mathbf{S} + 3\mathbf{F} + 3\mathbf{E} + \mathbf{V}}{8}$$

- where S is the average of the solid points for each of the hexahedra that have this as a vertex, $\mathbf{F}$ is the average of the face points for the faces that contain the vertex, $\mathbf{E}$ is the average of the edge points for the edges that radiate from this vertex, and $\mathbf{V}$ is the vertex itself.

6. Catmull-Clark Solids

To extend the rules developed for trivariate cubic uniform B-spline solids to lattices of arbitrary topology is straightforward. Again, we can classify the refinement points into four types:

- solid points – the points are the average of the control points in the lattice that bound the cell containing this point.
- face points – the points can be written as

  $$\frac{\mathbf{S}_1 + \mathbf{S}_2 + 2\mathbf{F}}{4}$$

  where $\mathbf{S}_1$ and $\mathbf{S}_2$ are the solid points of the two cells that contain the face and $\mathbf{F}$ is the face point calculated by the averaging the control points that form the face.
- edge points – the points can be written as

  $$\frac{\mathbf{S} + 2\mathbf{F} + \mathbf{E}}{4}$$

  where S is the average of the solid points for those cells that contain the edge, $\mathbf{F}$ is the average of the face points (calculated as the average of the control points forming the face) for those faces that contain the edge, and $\mathbf{E}$ is the midpoint of the edge.
- vertex points – the points can be written as

  $$\frac{\mathbf{S} + 3\mathbf{F} + 3\mathbf{E} + \mathbf{V}}{8}$$

- where S is the average of the solid points for each of the cells that contain the vertex, $\mathbf{F}$ is the average of the face points (average of the vertices that surround a face) for the faces that contain the vertex, $\mathbf{E}$ is the average of the edge points (midpoints of the edges) for the edges that radiate from this vertex, and $\mathbf{V}$ is the vertex itself.

7. Conclusion

We have developed the refinement rules for Catmull-Clark solids. In this effort, we have tried to be uniform and complete in the derivation of the rules – starting with the B-spline univariate, bivariate and trivariate cases and developing the general cases in a straightforward way.

We have not discussed continuity conditions with these solids, and have not extended the Catmull-Clark extensions for vertex points in the surface case to the solid case. This is a topic for another paper.

Much is yet to be done with these solids. They form a new paradigm of a solid, defined by a lattice of points with an arbitrary connection topology, and they need to be examined further.

References

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

[2] Chaikin, G. An algorithm for high speed curve generation. *Computer Graphics and Image Processing 3* (1974), 346–349.

[3] Doo, D. A subdivision algorithm for smoothing down irregularly shaped polyhedrons. In *Proceedings of the Int'l Conf. Interactive Techniques in Computer Aided Design* (1978), pp. 157–165.

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

[5] Halstead, M., Kass, M., and DeRose, T. Efficient, fair interpolation using Catmull-Clark surfaces. In *Computer Graphics (SIGGRAPH '93 Proceedings)* (Aug. 1993), vol. 27, pp. 35–44.

[6] Hoppe, H., DeRose, T., Duchamp, T., Halstead, M., Jin, H., McDonald, J., Schweitzer, J., and Stuetzle, W. Piecewise smooth surface reconstruction. In *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, Computer Graphics Proceedings, Annual Conference Series, pp. 295–302.

[7] Lounsbery, M. *Multiresolution Analysis for Surfaces of Arbitrary Topological Type.* PhD thesis, Department of Computer Science and Engineering, University of Washington, Seattle, WA, June 1994.

[8] Peters, J. Smooth free-form surfaces over irregular meshes generalizing quadratic splines. *Computer Aided Geometric Design 10*, 3 (Aug. 1993), 347–362.

[9] Riesenfeld, R. On Chaikin's algorithm. *Computer Graphics and Image Processing 4*, 3 (1975), 304–310.

[10] Sederberg, T. W., and Parry, S. R. Free-form deformation of solid geometric models. In *Computer Graphics (SIGGRAPH '86 Proceedings)* (Aug. 1986), vol. 20, pp. 151–160.