Spline-based Solid Subdivision Schemes over Arbitrary Tetrahedral Meshes

Yu-Sung Chang *, Hong Qin

Department of Computer Science
State University of New York at Stony Brook
Stony Brook, NY 11794–4400

Abstract

In this paper, we propose novel trivariate spline-based solid subdivision schemes over arbitrary tetrahedral meshes in 3D space. The motivation of the proposed solid subdivision schemes are to represent solid objects with complex topology and heterogeneous material properties. The subdivision schemes are based on trivariate double-directional box splines which serve as their basis functions. By devising a special quasi-regular structure comprising tetrahedra and octahedra in 3D, we derive the subdivision rules for the regular cases. The subdivision algorithm uses a certain directional choice of the diagonals in octahedra to precisely evaluate the basis functions in the limit. We extend the subdivision rules to cope with arbitrary tetrahedral meshes. To avoid the asymmetry of the original scheme, an additional subdivision scheme based on averaging is presented. We prove the C^1 continuity of our subdivision scheme using existing mathematical techniques, such as the spectral analysis of subdivision matrices and the characteristic map method. Both theoretical and numerical results are presented in detail. Moveover, we support our hypothetical assumptions with empirical data which cover most of the practical cases. With the outline of the implementation of the algorithm, we present some experimental modeling results using our subdivision scheme, including solid models with non-trivial topology, volumetric objects with heterogeneous materials, and simple numerical simulation on solid objects.

Key words: Spline representations, subdivision algorithms, geometric and topological representations, solid modeling, multiresolution models, volumetric meshes.

* Corresponding author.
Email addresses: yusung@cs.sunysb.edu (Yu-Sung Chang), qin@cs.sunysb.edu (Hong Qin).

1 Introduction

Solid modeling, especially representing volumetric objects have been an important issue since the beginning of computer-aided design. Intuitively, representing geometric models as solids are more natural approach, since real-world and manufactured objects always have internal structure, unlike curves or surfaces that are usually used to display objects in 2D computer screen. The contemporary solid representations in solid modeling can be categorized as the followings: Implicit function representations such as Constructive Solid Geometry (CSG) and blobby models, boundary representations (B-reps), parametric representations, polyhedral mesh representations and cell decomposition (voxels). Each representation has its own benefits and limitations. In fact, the solid representations themselves have not been changed fundamentally since the famous solid modeling survey by Requicha and Voelcker (1982). However, the recent emergence of new technologies poses new challenges to the existing solid representations. One such example is stereolithography, or 3D laying technology used in rapid prototyping process. This technology involves the thin-layering of 3D models, the injection process of special liquid photopolymer, and the hardening process using the stereolithograph apparatus (SLA). The technology can produce a real object of an arbitrary solid model using the liquid plastic. Since the process requires the continuous layering of the inside of 3D models, the ideal computer models for this process should be able to represent continuous varying material properties inside of the models, as well as boundaries. Functionally graded materials (FGM) (Ghosh et al., 1997; Miyamoto et al., 1999) are another examples of the materials with continuously varying properties. These type of materials, often referred as heterogeneous, or anisotropic materials are emerging rapidly in many practical fields. On the contrary to the necessity in these technologies, the underlying assumption of the most of the existing representations, especially CSG and B-reps, is that the material is homogeneous. This fact makes many existing solid representations very inadequate for representing the new materials. Another challenge associated with new technologies are the growing size of the volumetric data. When the first magnetic resonance imaging (MRI) system was introduced, the resolution was very limited. However, the cutting edge 3 Tesla MRI machines (General Electric Company, 2004) can achieve the resolution as high as 1 mm^3 per voxel. As a consequence, they can generate very large volumetric data in a short period of time. In addition, the data from the other sources, such as the seismic data from geo-scientific survey, various medical images, and the point clouds from 3D scanning machines increase the amounts and the complexity of the volume data that should be dealt by the solid modeling techniques. Moreover, new paradigms in shape design, such as the free-form design (McDonnell and Qin, 2000), the interactive design, the physics-based modeling, and the integration of shape modeling and engineering analysis (Zagajac, 1996) introduce new challenges to the existing solid representations. We can briefly summarize the problems as follows:

(a) The growth of geometric and topological complexities
(b) The emergence of heterogeneous materials
(c) Vary large volumetric data
(d) New modeling paradigms.

To resolve these new issues in solid modeling, we propose new solid representation based on trivariate box splines and subdivision algorithms. We develop our representation over arbitrary tetrahedral meshes in 3D. We introduce two new solid subdivision schemes based on trivariate box splines. In addition, we provide mathematical analysis to guarantee a certain level of continuity. We begin with a brief history on subdivision method, especially associated with solid modeling.

1.1 Subdivision Method

Subdivision method in computer graphics involves successive refinement of initial control points to acquire smooth geometric objects in the limit. The one of the earliest and the most simple subdivision methods related to computer graphics can be found in Chaikin's algorithm (Chaikin, 1974) to represent a spline curve. After few years, Catmull and Clark (1978) and Doo and Sabin (1978) simultaneously published remarkable papers regarding subdivision surfaces. Since then, much research related to subdivision surfaces (Loop, 1987; Dyn et al., 1990; Hoppe et al., 1994; Prautzsch and Umlauf, 1998; Kobbelt, 1996b,a; Levin, 1999) and their analysis (Prautzsch, 1985; Zorin et al., 1996; Zorin, 2000; Levin and Levin, 2003) has been presented in the past decades. In fact, the subdivision surfaces have gained popularity in computer graphics, especially in computer animation, mainly due to the following advantages over the other surface representations:

(a) Uniformity of representation
(b) Multiresolution analysis and level-of-detail control
(c) Numerical efficiency and stability
(d) Handling of arbitrary topology or genus
(e) Simplicity in implementation
(f) Visual quality.

Despite of these advantages which can obviously benefit solid modeling as well as surface case, most of the previous work has been focused on surfaces, rather than solids. There have been few attempts to apply the subdivision method to solid representation. One example is the work by MacCracken and Joy (1996), which generalizes tri-cubic B-splines to solids of arbitrary topology. In this approach, they used the tensor-product of the Catmull-Clark surface scheme to represent solids. Inherently, underlying meshes of their objects must be hexahedral. They employed their scheme mainly for free-form deformation of existing triangular models. Later on, Bajaj et al. (2002) further extended the scheme with the analysis based on numerical experiments. They approached the problem by separating the process into multi-linear subdivision followed by a cell averaging operation. Therefore, they call it MLCA or Multi-linear Cell Averaging scheme. Since the scheme makes no assumption on the local topology of the hexahedral mesh, it can be directly applied to non-manifold situations. The continuity analysis that they represented are mostly based on numerical experiments. Linsen et al. (2002) proposed $\sqrt[4]{2}$ subdivision to represent time-varying volume data in hierarchical fashion. This four-dimensional subdivision scheme provides both spatial and temporal scalability. Also, they used quadri-linear B-spline wavelet lifting scheme based on the $\sqrt[4]{2}$ subdivision for high-quality data approximation. Pascucci (2002)'s recent work on high dimensional subdivision scheme suggested interesting cell-split of arbitrary polyhedra to slow the increase of cells during the subdivision process. It introduces special *diamond* cells during the process, which makes it hard to analyze. Chang et al. (2002, 2003) proposed an approximate and interpolatory schemes based on tetrahedral meshes. Unlike the other solid subdivision schemes, their approaches employ non-tensor-product trivariate functions, which enable the usage of arbitrary tetrahedral meshes, instead of hexahedral meshes. The most recent subdivision scheme proposed by Schaefer et al. (2004) is based on the quasi-regular structure that has been proposed by Chang et al. (2002) and their paper contains detailed analysis on special cases, especially, the face-to-face case, using the joint spectral analysis by Levin and Levin (2003). One shortfall of their approach is lack of explicit basis functions even in regular cases, since it is not based on any splines unlike Chang et al.'s approach.

1.2 Contribution

In this paper, we propose new solid subdivision schemes to address the issues entailed by technological advances. Our contribution in this research can be summarized as the followings:

- We propose quasi-regular structure based on tetrahedra and octahedra. We prove that the proposed mesh satisfies the properties that are desired for the subdivision method. This provides the foundation of the derivation of subdivision solid schemes over arbitrary tetrahedral meshes. Tetrahedral meshes provide our schemes simplicity and flexibility that can resolve the geometric and topological complexities.
- We derive new approximate solid subdivision schemes from the recursive property of box splines. Based on a particular class of trivariate box splines, we achieve high smoothness without the limitation of tensor-product subdivision schemes. We generalize the schemes to arbitrary tetrahedral meshes in 3D. The schemes intrinsically share the advantages of the subdivision methods, such as the mutiresolution structure and level-of-detail control which are very desirable for large volumetric data processing.

- We employ current mathematical tools for the convergence and continuity analysis of the schemes. We discuss the new challenges associated with the solid scheme analysis and apply the analysis techniques to a certain extend. The empirical data support the continuous property of objects represented by our schemes. The continuity analysis guarantees that our schemes are appropriate to represent heterogeneous solid objects, especially continuously varying materials.

We also address the key issues in implementation and offer some experimental examples, such as solid models with non-trivial topology, volumetric objects heterogeneous materials, and simple numerical simulation on solid objects.

2 Box Splines Revisited

There has been much study on box splines since the early days of computer-aided design. We briefly discuss the general definition and properties of the box splines in this section for reminder, which are essential for the development of new subdivision schemes. Furthermore, we introduce the concept of generating functions and discrete convolution.

2.1 Definition

There are several ways to define the box spline. One constructive way is by considering a projective image of the n-dimensional unit cube, or n*-cube*, to m-dimensional space with respect to an $m \times n$ projection matrix Ξ (Boehm, 1984). We will follow the notations and the definition in *the* box spline book (de Boor et al., 1993) by de Boor et al.Analytically, we define the box spline M_Ξ associated with the matrix Ξ as a distribution given by:

$$M_\Xi : \varphi \mapsto \langle M_\Xi, \varphi \rangle := \int_{\sqcup} \varphi(\Xi \mathbf{t}) d\mathbf{t}, \tag{2.1}$$

where $\varphi \in C(\mathbb{R}^m)$ and $\sqcup = [0,1)^n$, the half-open unit n-cube. By decomposing $\mathbb{R}^n = (\ker \Xi) \bigoplus (\ker \Xi)^\perp$ and applying Fubini's Theorem, we can derive that the distribution can be expressed as:

$$\langle M_\Xi, \varphi \rangle := \int_{\operatorname{ran} \Xi} \varphi(\mathbf{x}) \operatorname{vol}_{n-d} \left(\Xi^{-1}\{\mathbf{x}\} \cap \sqcup \right) d\mathbf{x} / |\det \Xi|, \tag{2.2}$$

where $\operatorname{ran} \Xi = (\ker \Xi)^\perp$ and $d = \dim(\operatorname{ran} \Xi)$. We identify the M_Ξ with a function:

$$M_\Xi(\mathbf{x}) = \operatorname{vol}_{n-d} \left(\Xi^{-1}\{\mathbf{x}\} \cap \sqcup \right) / |\det \Xi|, \tag{2.3}$$

when $\mathbf{x} \in \operatorname{ran} \Xi$. At each point $\mathbf{x}$, the value $M_\Xi(\mathbf{x})$ is defined by the volume of the cross-section area of $\sqcup$ with $\Xi^{-1}\{\mathbf{x}\}$, divided by the volume of the projected image of a unit volume in $\operatorname{ran}\Xi$. This intrinsic interpretation is important in understanding the subdivision process. Finally, a geometric object S in $\mathbb{R}^m$ can be represented by the sum of the box splines:

$$S(\mathbf{x}) = \sum_{\mathbf{i}\in\mathbb{Z}^m} \mathbf{p}_\mathbf{i} M_\Xi(\mathbf{x}-\mathbf{i}), \tag{2.4}$$

where $\mathbf{p}_\mathbf{i}$ are initial control points in $\mathbb{R}^m$.

The analytic definition leads us to a very useful inductive definition. When Ξ is invertible (i.e. $n = m = d$), it is clear that M_Ξ is the normalized characteristic function of $\Xi\square$:

$$M_\Xi = \frac{1}{|\det \Xi|}\chi_{\Xi\square}. \tag{2.5}$$

In addition, if $\Xi \cup \zeta$ is a matrix formed by the addition of the column $\zeta \in \mathbb{R}^m$ to the matrix Ξ, the box spline $M_{\Xi\cup\zeta}$ is given by the convolution equation:

$$M_{\Xi\cup\zeta}(\mathbf{x}) = \int_0^1 M_\Xi(\mathbf{x}-t\zeta)dt, \tag{2.6}$$

or simply,

$$M_{\Xi\cup\zeta} = M_\Xi * M_\zeta. \tag{2.7}$$

2.2 Properties

Generally, the box spline satisfy the following properties:

(a) *Positive definition*: $M_\Xi \geq 0$ and $\int_{\operatorname{ran}\Xi} M_\Xi = 1$.
(b) *Partition of unity*: $f(\mathbf{x}) = \sum_{\mathbf{i}\in\mathbb{Z}^m} M_\Xi(\mathbf{x}-\mathbf{i}) = 1$.
(c) *Piecewise polynomial*: M_Ξ is a piecewise polynomial of degree $n-m$.
(d) *Continuity*: M_Ξ is a $C^{n-\widetilde{n}-2}$ function, where $\widetilde{n}$ is the maximal number of columns of Ξ that does not span $\mathbb{R}^m$.

We refer the readers who are interested in the proofs of the properties to the box spline book (de Boor et al., 1993) by de Boor et al.. In addition to the properties, a box spline can be expressed by linear combinations of integer shifts of the box splines with the support of a half-size. It can be formulated as:

$$M_\Xi(\mathbf{x}) = \sum_{\mathbf{i}\in\mathbb{Z}^m} s_\mathbf{i} M_\Xi(2\mathbf{x}-\mathbf{i}). \tag{2.8}$$

The $s_\mathbf{i} \in \mathbb{R}$ is called a *subdivision* coefficient, and the formula is called a *subdivision algorithm*.

The simplest case is when $\Xi = \mathbf{E}$, the $m \times m$ identity matrix. It is obvious that $M_{\mathbf{E}} = \chi_{\mathbf{E}\Box}$, and it has a simple subdivision formula:

$$M_{\mathbf{E}}(\mathbf{x}) = \sum_{\mathbf{i}\in\mathbb{Z}^m} M_{\mathbf{E}}(2\mathbf{x} - \mathbf{i}). \tag{2.9}$$

We can easily derive subdivision formulae for the other box splines by using the convolution equation (2.6) and the following theorem.

Theorem 2.1. *Suppose M_Ξ has a subdivision formula $M_\Xi = \sum_{\mathbf{i}} s_{\mathbf{i}} M_\Xi(2\mathbf{x} - \mathbf{i})$ and $\Phi = \Xi \cup \zeta$, where ζ is a column vector in $\mathbb{Z}^m$. Then M_Φ satisfies the following subdivision formula:*

$$M_\Phi(\mathbf{x}) = \frac{1}{2}\sum_{\mathbf{i}}(s_{\mathbf{i}} + s_{\mathbf{i}-\zeta})M_\Phi(2\mathbf{x} - \mathbf{i}). \tag{2.10}$$

Proof. By the convolution formula (2.6),

$$\begin{aligned}
M_\Phi(\mathbf{x}) &= \int_0^1 M_\Xi(\mathbf{x} - \mathbf{t}\zeta)d\mathbf{t} \\
&= \int_0^1 \sum_{\mathbf{i}} s_{\mathbf{i}} M_\Xi(2(\mathbf{x} - \mathbf{t}\zeta) - \mathbf{i})d\mathbf{t} \\
&= \frac{1}{2}\sum_{\mathbf{i}} s_{\mathbf{i}} \int_0^2 M_\Xi(2\mathbf{x} - \mathbf{i} - \mathbf{u}\zeta)d\mathbf{u} \\
&= \frac{1}{2}\sum_{\mathbf{i}} \left[s_{\mathbf{i}} \int_0^1 M_\Xi(2\mathbf{x} - \mathbf{i} - \mathbf{u}\zeta)d\mathbf{u} + s_{\mathbf{i}} \int_1^2 M_\Xi(2\mathbf{x} - \mathbf{i} - \mathbf{u}\zeta)d\mathbf{u} \right] \\
&= \frac{1}{2}\sum_{\mathbf{i}} \left[s_{\mathbf{i}} M_\Phi(2\mathbf{x} - \mathbf{i}) + s_{\mathbf{i}} \int_0^1 M_\Xi(2\mathbf{x} - \mathbf{i} - \zeta - \mathbf{u}\zeta)d\mathbf{u} \right] \\
&= \frac{1}{2}\sum_{\mathbf{i}} \left[s_{\mathbf{i}} M_\Phi(2\mathbf{x} - \mathbf{i}) + s_{\mathbf{i}} M_\Phi(2\mathbf{x} - \mathbf{i} - \zeta) \right] \\
&= \frac{1}{2}\sum_{\mathbf{i}}(s_{\mathbf{i}} + s_{\mathbf{i}-\zeta})M_\Phi(2\mathbf{x} - \mathbf{i}).
\end{aligned}$$

$\square$

Even though it is possible to derive the subdivision algorithm for a particular box spline by Theorem 2.1, it is more convenient to inspect the coefficients themselves, as a discrete convolution (Warren and Weimer, 2001). We explain the method in the next section.

A subdivision algorithm can be considered as the discrete convolution among polynomials (Warren and Weimer, 2001; Dyn and Micchelli, 1990). Once the relation between the subdivision coefficients and the polynomials (usually called *generating function*) are established, the subdivision formula can be acquired systematically. Here, we briefly discuss the most important property of the generating function method related to the subdivision process.

We associate a subdivision formula with a multivariate polynomial. Suppose M_Ξ has a subdivision formula $M_\Xi = \sum_{\mathbf{i}} s_{\mathbf{i}} M_\Xi(2 \cdot - \mathbf{i})$, then we assign:

$$f_\Xi(\mathbf{z}) = \sum_{\mathbf{i} \in \mathbb{Z}^m} s_{\mathbf{i}} \mathbf{z}^{\mathbf{i}}, \tag{2.11}$$

as its generating function, where $\mathbf{z} = (z_1, \cdots, z_m)$ and $\mathbf{z}^{\mathbf{i}} = ({z_1}^{i_1}, \ldots, {z_m}^{i_m})$. It is clear to see that one can derive the analog of Theorem 2.1 for the generating function, as following:

Corollary 2.2. *Suppose M_Ξ has a subdivision formula associated with a generating function $f_\Xi(\mathbf{z}) = \sum_{\mathbf{i} \in \mathbb{Z}^m} s_{\mathbf{i}} \mathbf{z}^{\mathbf{i}}$. If $\Phi = \Xi \cup \zeta$, then M_Φ has a subdivision formula whose generating function is:*

$$f_\Phi(\mathbf{z}) = \frac{1}{2} \sum_{\mathbf{i} \in \mathbb{Z}^m} (s_{\mathbf{i}} + s_{\mathbf{i}-\zeta}) \mathbf{z}^{\mathbf{i}}. \tag{2.12}$$

The result of Corollary 2.2 can be rewritten as $f_\Phi(\mathbf{z}) = \frac{1}{2}(1+\mathbf{z}^\zeta) f_\Xi(\mathbf{z})$. By applying Corollary 2.2 recursively, we can acquire the following general formula for the box splines.

Corollary 2.3. *For any box spline M_Ξ represented by the directional matrix Ξ, its generating function f_Ξ can be written as:*

$$f_\Xi(\mathbf{z}) = \frac{1}{2^{d-k}} \prod_{\zeta} (1 + \mathbf{z}^\zeta). \tag{2.13}$$

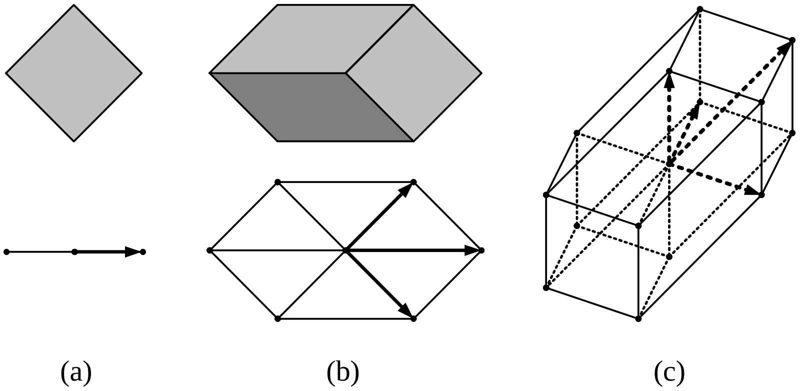
Here, d is the number of columns in the matrix Ξ and ζ is each column vector of Ξ. The proofs are straightforward, therefore they are omitted. For further discussion on generation functions, we refer readers to the book by Warren and Weimer (2001).

3 Derivation of Subdivision Scheme

We have defined general box splines and demonstrated their properties in the previous section. Moreover, we have established a subdivision algorithm for the box splines, which leads us to the derivation of subdivision schemes. In this section, we focus on a particular type of box splines of our interest, namely, the *double directional* box splines. We discuss the subdivision rules to evaluate the box splines and extend them into general situations, where a mesh consists of arbitrary tetrahedra in 3D. In Appendix A, we propose an additional subdivision scheme based on spatial averaging to avoid asymmetry inherited in the original subdivision scheme.

3.1 Double Directional Box Splines

We choose *double directional* box splines and their variations to be our choice of the spline representation in regular cases. First, we begin with considering a piecewise linear box spline function. Suppose $\Xi_1 = \mathbf{E} \cup \delta$, where $\delta = [1, \cdots, 1] \in \mathbb{R}^m$. Then, M_{Ξ_1} forms a piecewise linear function over $\mathbb{R}^m$. The support of the spline function is the form of two unit squares, or m-cubes, sharing one vertex, with additional edges that join the corresponding vertices in each cube. Figure 1 shows the supports of the piecewise linear box splines for $m = 1$, 2, and 3,respectively. The subdivision algorithm for the piecewise linear box splines can be derived intuitively. As shown in 2, each n-cube $\square \subset \mathbb{R}^n$ can be split into 2^n sub-cubes. The projected images of the sub-cubes by Ξ_1 constitute the piecewise linear box splines with half-support. This procedure admits a simple subdivision algorithm for each dimension.



(a) (b) (c)

Fig. 1. The supports of the piecewise linear box splines.

The double directional box spline can be understood as a projection of the unit $2(m+1)$-cube into $\mathbb{R}^m$, as previously mentioned. The spline can be defined by the directional matrix $\Xi_2 = \Xi_1 \cup \Xi_1$. We use the union operation to represent the conjoined matrix. Since the matrix Ξ_2 projects a pair of edges into the same place in $\mathbb{R}^m$, the supports of the double directional box splines are the exactly same shapes as those of the piecewise linear box splines. Note that, for $m = 1$ and 2, the supports of the box splines are always m-simplex, *i.e.*, line segments and triangles. Moreover, the support of the 2D case comprises a regularly structured mesh in $\mathbb{R}^2$. Since our objective is to find a subdivision algorithm over arbitrary tetrahedral meshes in 3D, these results seem promising. However, as we will discuss in the next section, this is non-trivial question in $\mathbb{R}^3$.

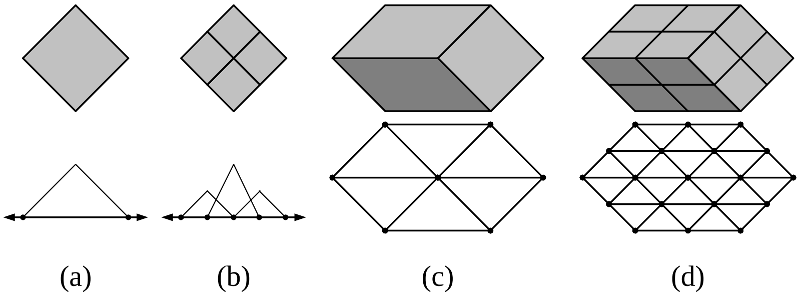


Fig. 2. An intuitive subdivision of the piecewise linear box splines.

3.2 Regular Subdivision Meshes

During any subdivision process, we introduce new vertices associated existing geometric elements – vertices, edges, faces, and cells for solid cases – using the information about their neighbors and the weight values derived from the subdivision scheme. The neighbor information is acquired from the connectivity among the vertices, which is given by the form of *subdivision meshes*. Generally speaking, subdivision meshes are undirected graphs in $\mathbb{R}^m$. The subdivision meshes are essential in any subdivision scheme. Not only they provide the connectivity information during the process, but also they are deeply connected with the derivation and the generalization of the scheme. In particular, it is desired, but not necessary, for any subdivision meshes of stationary subdivision schemes to satisfy the following properties:

(a) *Cell decomposition*: A subdivision mesh consists of polygonal or polyhedral cells of dimension m in $\mathbb{R}^m$.
(b) *Self-congruency*: Each cell, if divided by a proper split method, yields subcells which are congruent to the parent cell.

It is possible that the mesh does not yield a valid polyhedral decomposition in $\mathbb{R}^m$. However, it becomes complicated to generalize the subdivision schemes to arbitrary meshes. In the ideal case, the subdivision algorithm will be infinitely repeated to acquire the limit object. If we consider the infinite splits of each cell, it is obvious that any cells that satisfy the self-congruency can tile the entire $\mathbb{R}^m$ space. On the other hand, it is trivial that a part of any regular structured mesh, or a *tiling* in $\mathbb{R}^m$ satisfies the properties. The understanding of regular structured meshes in $\mathbb{R}^m$

is essential to understand the subdivision process. As mentioned in the previous section, the support of the box splines in the 2D case provides a part of such regular mesh. In particular, this is a regular triangle mesh with the valence 6. For the 3D case, the boundary of the support of the box splines (Figure 1(c)) forms a rhombic dodecahedron, which is a known tiling polyhedron in $\mathbb{R}^3$. However, if you consider the other edges, the support does not admit any proper polyhedral decomposition.

To continue investigating structured meshes in high dimensional spaces, we adopt Schläfli symbol to represent regular structures (Coxeter, 1963). The Schläfli symbol is defined as the following:

Definition 3.1. The Schläfli symbol $\{p\}$ defines a regular p-gon. The Schläfli symbol $\{p, q\}$ represents a regular structured mesh in $\mathbb{R}^2$, such that each cell is the regular p-gon and there are q of them at each vertex. The Schläfli symbol $\left\{ {p \atop q} \right\}$ represents a dual quasi-regular structured mesh of $\{p, q\}$. In 3D space, the Schläfli symbol $\{p, q, r\}$ defines a regular structured mesh in $\mathbb{R}^3$, such that each cell is $\{p, q\}$, *i.e.*it is a regular polytope with the faces of the regular p-gon and there are q of them at each vertex, and there are r of them surrounding each edge. The same analogue holds for higher dimensional structures.

By definition, any regular structure consists of a single type of a polyhedron. In addition, there exist structures that are less regular, such as *quasi-*, or *semi-regular* structures. To understand *quasi-regular* structure, we first need to introduce *vertex figures*. The vertex figure of a vertex of a structured mesh is defined as a polyhedron whose vertices are the mid-points of all the edges that emanate from the given vertex. A polyhedron is called quasi-regular, if its faces are regular and the vertex figure is cyclic and equiangular. A mesh is called quasi-regular if its cells are regular while its vertex figures are quasi-regular.

The Schläfli symbols offer a convenient way to deal with structured meshes. For example, the triangular structured mesh in $\mathbb{R}^2$ by the 2D support (Figure 1(b)) can be written as $\{3, 6\}$ by the Schläfli symbol. By examining the properties of the structured meshes using the Schläfli symbols, one can prove that for every $m \geq 2$, $\{4, 3^{m-2}, 4\}$ forms a regular structured mesh in $\mathbb{R}^m$. In fact, this is nothing more than the mesh equivalent to the lattice structure $\mathbb{Z}^m$. Unfortunately, it is the only regular structure for $m = 3$ and $m \geq 5$. For $m = 4$, there exist two more regular structured meshes, $\{3, 3, 4, 3\}$ and the reciprocal of it, $\{3, 4, 3, 3\}$.

Finding structured meshes in 3D involving tetrahedra is more difficult than the triangle case. We have showed that the only regular structure in $\mathbb{R}^3$ is $\{4, 3, 4\}$. Even without the Schläfli symbols, the problems are apparent. Figure 3 illustrates some examples of splits of a tetrahedron. As we can see, none of them yields congruent sub-tetrahedra. In fact, there is no tetrahedral split that produces congruent subcells. Moreover, the two splits except the edge-split produce slimmer tetrahedra and increase the valences of vertices. These behaviors are inappropriate for subdi-

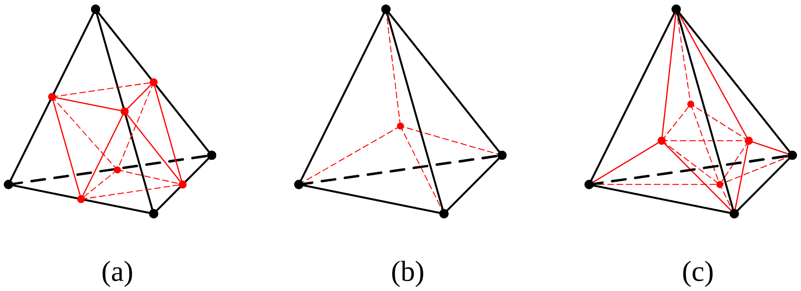


Fig. 3. Examples of splits of a tetrahedron. (a) The edge-split. (b) The cell-centroid-split. (c) The face-centroid-split.

vision schemes. Actually, the edge-split does yield 4 congruent tetrahedra (Figure 4(a)). However, it generates an octahedra in the middle. If we keep splitting the octahedron by the same method (Figure 4(b)) and repeat the process, we result in the structure called *octet-truss* (Figure 4(c)). In fact, this is the only quasi-regular structure in $\mathbb{R}^3$, represented by the Schläfli symbol $\{3, {3 \atop 4}\}$. As the symbol suggests, it consists of two types of polyhedra, *i.e.*tetrahedra ($\{3, 3\}$) and octahedra ($\{3, 4\}$).

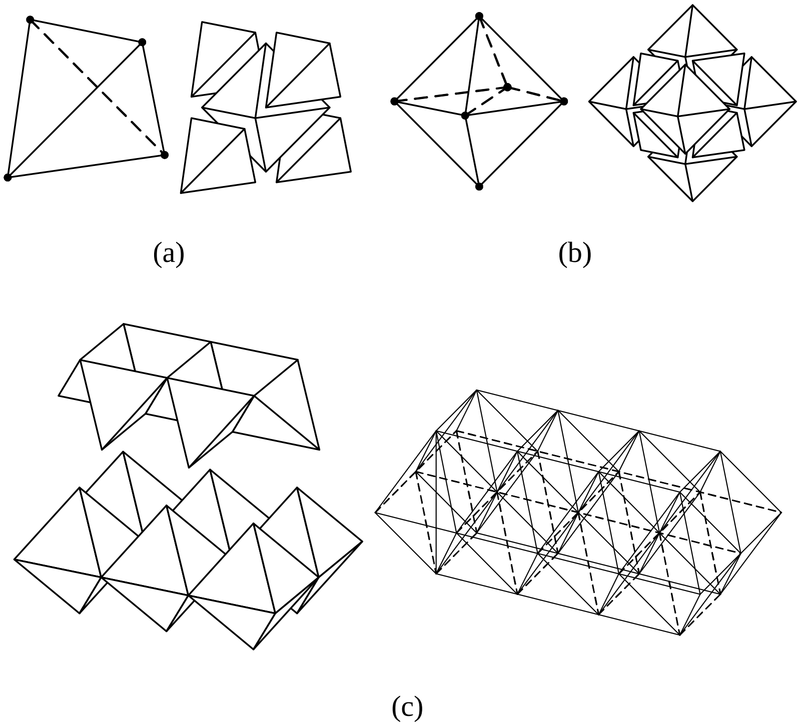
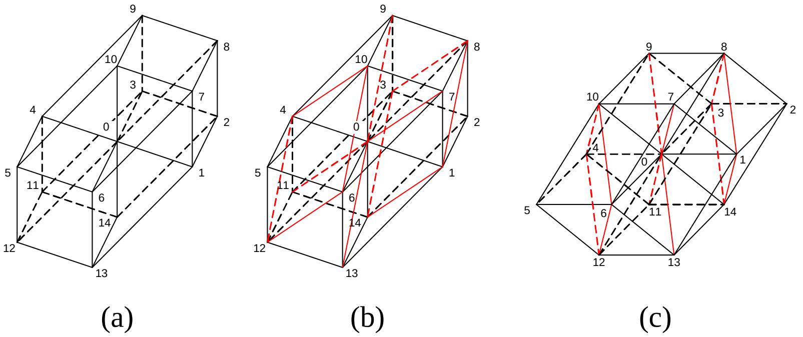


Fig. 4. The octet-truss structure. (a) The edge-split of a tetrahedron generates 4 tetrahedra and 1 octahedron. (b) The edge-split of an octahedron generates 8 tetrahedra and 6 octahedra. (c) By combining these two, we form a quasi-regular structured mesh, called octet-truss.

It is easy to confirm that the octet-truss structured mesh satisfies the subdivision mesh properties. In the rest of the paper, the octet-truss structured mesh will serve as our underlying mesh for the subdivision scheme. This is no arbitrary choice, as illustrated in Figure 5. From the support of the trivariate piecewise linear box spline by Ξ_1 (Figure 5(a)), we can add minimal number of edges to provide a proper polyhedral decomposition (Figure 5). An easy one-to-one correspondence with a part of the octet-truss mesh can be easily established by the inspection of Figure 6. Since the support of the trivariate double directional box spline by Ξ_2 is the same as that of the piecewise linear case, there is no difficulty to derive the subdivision rules for the trivariate double directional box spline over the structured mesh.



(a) (b) (c)

Fig. 5. (a) The projected image of the 4- and 8-cube by Ξ_1 and Ξ_2, respectively. (b) New edges (red) are added to make proper polyhedral decomposition. (c) The one-to-one correspondence with a part of the octet-truss mesh.

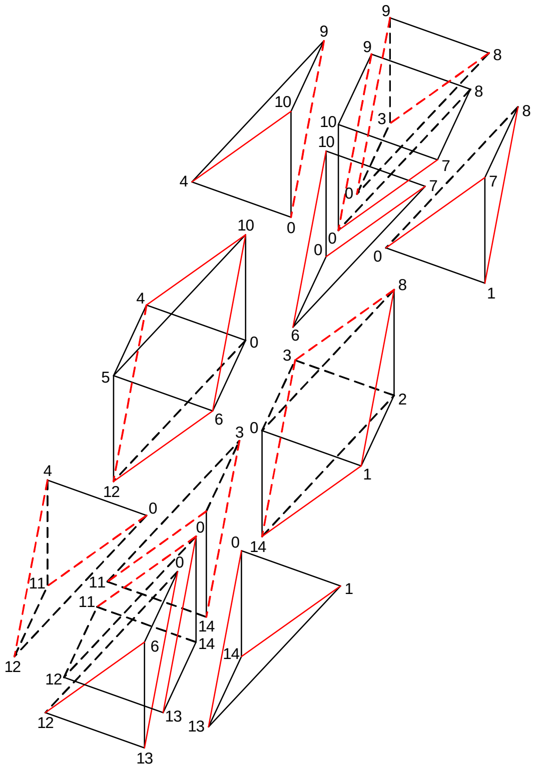


Fig. 6. The decomposition of the support of the box splines with the additional edges (red).

3.3 Generating Function over Subdivision Mashes

We are now concerned in deriving the subdivision rules for the trivariate box spline. There is a systematical way to derive the subdivision rules for the regular cases from the coefficients of generating functions. We proceed with the following steps:

(a) Compute the generating function of the subdivision algorithm.
(b) Associate the acquired subdivision coefficients from the generating function with $\mathbb{Z}^m$ by the power of their polynomial terms. For instance, if the coefficient $c_{\mathbf{i}}$ is for the term $\mathbf{z}^{\mathbf{i}} = ({z_1}^{i_1}, \cdots, {z_m}^{i_m})$, the $c_{\mathbf{i}}$ is assigned to the lattice point $\mathbf{i} = (i_1, \cdots, i_m)$.
(c) Index a part of the subdivision mesh properly so that we can establish the one-to-one correspondence between the non-zero coefficients and the mesh vertices.
(d) Extract the subdivision masks for the vertices, edges, and cells if necessary.
(e) Reduce the number of the masks by considering rotational symmetry.

We start with the bivariate double directional box spline as an example. By applying the directional matrix $\Xi = \{(1,0),(0,1),(1,1),(1,0),(0,1),(1,1)\}$ to Corollary 2.3, we compute the generating function:

$$f_{\Xi}(z_1, z_2) = \frac{1}{16}(1+z_1)^2(1+z_2)^2(1+z_1z_2)^2. \tag{3.1}$$

Its coefficients can be properly embedded in the 2-ring vertex neighbor of the $\{3,6\}$ structure (Figure 2(d)). From the equation (3.1), we derive the subdivision coefficients and place them on the part of $\mathbb{Z}^2$ as shown in Figure 7(a). The one-to-one correspondence between $\mathbb{Z}^2$ and the $\{3,6\}$ structure is canonical with the addition of the diagonal edges. Now we can extract the subdivision masks. For vertex cases, we consider the center vertex as our vertex point. Since we perform the subdivision using the edge-split, the edges of our masks should have the size of two grids in each direction. Therefore, we can extract the subdivision mask for regular vertices, shown in Figure 7(b). For edge cases, we consider the center as newly introduced edge point. Then, there are 3 choices of the length 2 edges and the associated masks, shown in Figure 7(c). Since all these masks are identical upon the rotation of the degree $\frac{2\pi}{3}$, we can reduce them into a single edge subdivision mask. In fact, the resulting rules are those of the Loop's scheme (Loop, 1987), since the Loop's scheme is based on the bivariate box spline.

The same procedure can be applied to the trivariate box spline case. By applying Corollary 2.3 to the directional matrix Ξ_2 in the 3D case, we acquire the following generating function:

$$f_{\Xi_2}(z_1, z_2, z_3) = \frac{1}{32}(1+z_1)^2(1+z_2)^2(1+z_3)^2(1+z_1z_2z_3)^2. \tag{3.2}$$

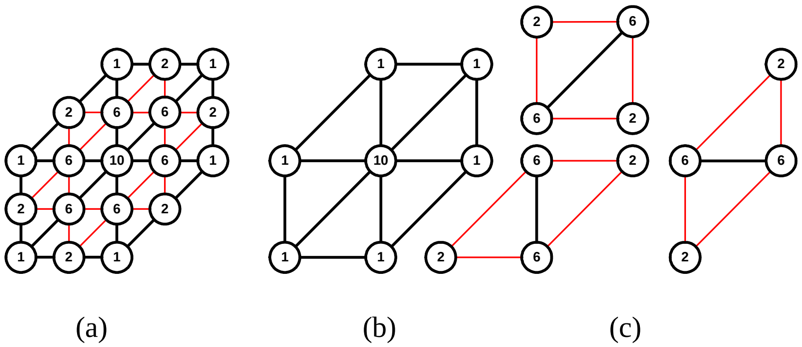


Fig. 7. (a) The subdivision coefficients of the Loop's scheme are embedded in the part of the $\{3,6\}$ regular structured mesh. (b) and (c) The extracted vertex and edge masks. All the values are to be multiplied by $\frac{1}{16}$.

The coefficient of each polynomial term $z_1^{i_1}z_2^{i_2}z_3^{i_3}$ represents the weight value for the subdivision algorithm at (i_1,i_2,i_3) in $\mathbb{Z}^3$. As explained in the previous section, by adding additional edges, we can establish the one-to-one correspondence between $\mathbb{Z}^3$ and the octet-truss. The coefficients of the formula (3.2) can be embedded in the part of $\mathbb{Z}^3$ corresponding to the 2-ring vertex neighbor of the octet-truss, as shown in Figure 8. The figure illustrates each layers of $\mathbb{Z}^3$ through z-axis. One must be cautious that there are edges between each layers as well as vertices with the same layer. In fact, all diagonals in Figure 8 represent the edges between layers, except the blue diagonals in the third layers. The mask extraction processes are shown in detail in Figure 9, Figure 10, and Figure 11. We follow the same steps as the 2D case, except that we now utilize the existence of the edges between layers. Unlike the 2D case, we have two types of the edge masks which are not identical by rotation. The edge masks (a-d) in Figure 10 have 6 neighbors in addition to each end point of the edge, whereas the edge mask (e-f) in Figure 10 have 4 neighbors. Each edge mask in (a-d) and in (e-f) is identified by rotation. Therefore, we have one vertex mask, two edge mask, and one cell mask for the subdivision algorithm.

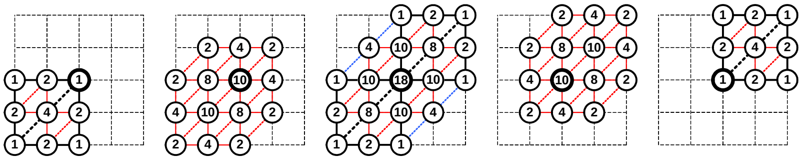


Fig. 8. The subdivision coefficients of the generating function are embedded in the part of the octet-truss regular structured mesh. All the values are to be multiplied by $\frac{1}{32}$. Since it is 3D mesh, each layer of the mesh is displayed separately. The bold circled vertex indicates the center of each layer.

This asymmetry in the edge masks occurs due to the unified direction choice of the diagonal in each octahedron, which are shown as blue edges in the third layers. Note that the original octet-truss mesh and its subdivision do not require the choice of the diagonal inside the octahedral cells. However, during the assignment of the subdivision coefficients onto the mesh, we need to choose one particular direction defined by one of the octahedral diagonals, because of the spherical asymmetry of

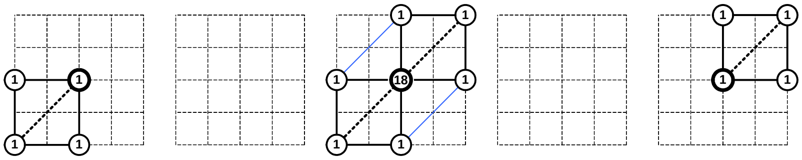
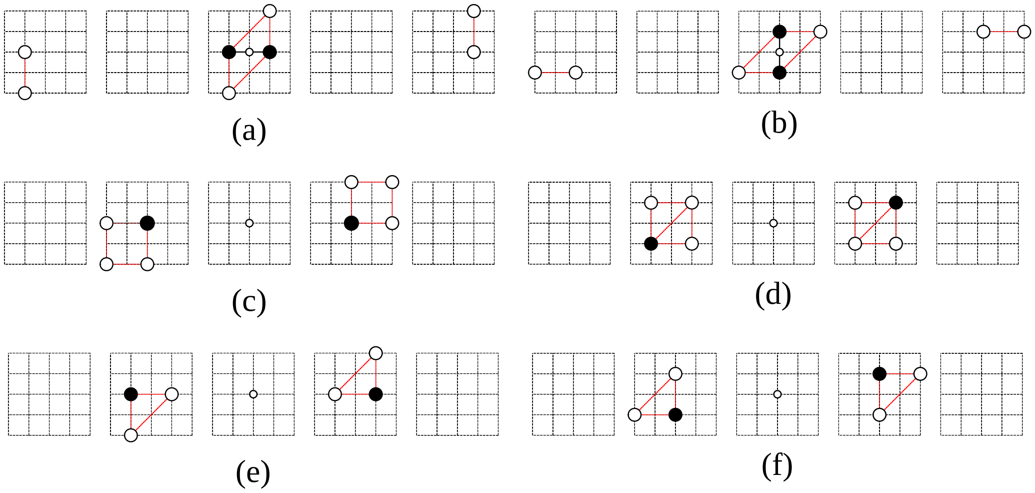


Fig. 9. The vertex mask for the box-spline based subdivision algorithm. All the values are to be multiplied by $\frac{1}{32}$.



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 10. The edge masks for the box-spline based subdivision algorithm. The small circles at the center represent new edge point. For (a-d), the subdivision coefficients are $\frac{2}{32}$ at the white circles and $\frac{10}{32}$ at the black circles. For (e-f), the values are $\frac{4}{32}$ and $\frac{8}{32}$, respectively.

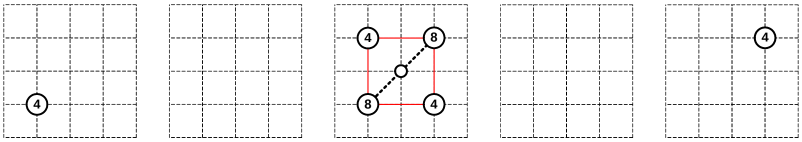


Fig. 11. The cell masks for the box-spline based subdivision scheme. All the values are to be multiplied by $\frac{1}{32}$. The small circle at the center represents new cell point.

the projected image of the 8-cube in 3D space.

We call this choice of the octahedral diagonal as the *major diagonal*. We have to choose the diagonal in such way that for two adjacent octahedra sharing an edge, the major diagonals are parallel to each other if the mesh is in canonical position. More precisely:

Definition 3.2. For each octahedral cell $\mathbf{o} = [\mathbf{x}_i, ..., \mathbf{x}_{i+5}]$ that comprises the mesh, we choose one pair of vertices $\{\mathbf{x}_{j_1}, \mathbf{x}_{j_2}\}$ which are not adjacent. The pair is called the *major diagonal* and is denoted by $\mu(\mathbf{o}) = [\mathbf{x}_{j_1}, \mathbf{x}_{j_2}]$. The major diagonal of a tetrahedral cell refers to the choice of major diagonal of the octahedron that occurs during the subdivision.

We call the choice of the major diagonals is *proper* if and only if, for two adjacent octahedra $\mathbf{o}_1 = [\mathbf{x}_1, \ldots, \mathbf{x}_6]$ and $\mathbf{o}_2 = [\mathbf{y}_1, \ldots, \mathbf{y}_6]$, each shared edge $[\mathbf{x}_{i_1}, \mathbf{x}_{i_2}] =$

$[\mathbf{y}_{j_1}, \mathbf{y}_{j_2}] \subset \mathbf{o}_1 \cap \mathbf{o}_2$ satisfies one of the following properties:

(a) $\mathbf{x}_{i_1} \in \mu(\mathbf{o}_1)$ and $\mathbf{x}_{i_2} \in \mu(\mathbf{o}_2)$,
(b) $\mathbf{x}_{i_2} \in \mu(\mathbf{o}_1)$ and $\mathbf{x}_{i_1} \in \mu(\mathbf{o}_2)$,
(c) Both $\mathbf{x}_{i_1}$ and $\mathbf{x}_{i_2}$ do not belong to any of $\mu(\mathbf{o}_i)$, $i = 1, 2$.

During the subdivision, sub-cells inherit the directions of the major diagonals from their parent cells. For a tetrahedral cell, the major diagonal direction is required to be stored because an octahedron is introduced during the subdivision. The proper choice of the major diagonals can be done easily in the regular cases. However, in general, the choice of the major diagonals causes a problem in the implementation of the subdivision algorithm. We will discuss more about the implementation issues in Section 5.

3.4 Regular Subdivision Rules

Since we have identified the subdivision masks, it is relatively easy to derive the regular subdivision rules for the subdivision algorithm. The rules reduce into one vertex rule, two edge rules and one cell rule. We begin with defining the neighbor set function $\rho(\cdot)$ of a vertex or an edge. We emphasize that the adjacency is defined by the existence of an edge or a major diagonal, not a cell. The formal definition of $\rho(\cdot)$ is as follows:

Definition 3.3. For each vertex $\mathbf{x}_i$, we say $\mathbf{x}_j \in \rho(\mathbf{x}_i)$ if and only if there exists an edge $\mathbf{e} = [\mathbf{x}_i, \mathbf{x}_j]$ or a major diagonal of an octahedral cell, $\mu(\mathbf{o}) = [\mathbf{x}_i, \mathbf{x}_j]$.

For each edge $\mathbf{e}_i = [\mathbf{x}_i, \mathbf{x}_{i+1}]$, we say $\mathbf{x}_j \in \rho(\mathbf{e}_i)$ if and only if there exists an edge $\mathbf{e} = [\mathbf{x}_k, \mathbf{x}_j]$ or a major diagonal of an octahedral cell, $\mu(\mathbf{o}) = [\mathbf{x}_k, \mathbf{x}_j]$ for both $k = 1, 2$.

3.4.1 Vertex Rule

Each regular vertex $\mathbf{x}_i$ has the valence of 14, *i.e.*$|\rho(\mathbf{x}_i)| = 14$. It is shared by 6 octahedra and 8 tetrahedra. However, because of the *proper* choice of the major diagonals and the definition of neighbors by Definition 3.3, we have only 14 adjacent vertices to be counted. Therefore, we introduce new vertex point $\mathbf{v}_{new}$ by:

$$\mathbf{v}_{new} = \frac{1}{32}\Big\{18\mathbf{x}_i + \sum_{\mathbf{x}_j \in \rho(\mathbf{x}_i)} \mathbf{x}_j\Big\}. \tag{3.3}$$

3.4.2 Edge Rules

Each edge $\mathbf{e}_i = [\mathbf{x}_i, \mathbf{x}_{i+1}]$ is shared by 2 octahedra and 2 tetrahedra. According to the position of the major diagonals of the octahedra, there are two types of the regular edge rules to compute new edge point $\mathbf{e}_{new}$ (Figure 12(b)). If $|\rho(\mathbf{e}_i)| = 6$ then,

$$\mathbf{e}_{new} = \frac{1}{32}\left\{10(\mathbf{x}_i + \mathbf{x}_{i+1}) + 2\sum_{\mathbf{x}_j \in \rho(\mathbf{e}_i)} \mathbf{x}_j\right\}. \tag{3.4}$$

If $|\rho(\mathbf{e}_i)| = 4$ then,

$$\mathbf{e}_{new} = \frac{1}{32}\left\{8(\mathbf{x}_i + \mathbf{x}_{i+1}) + 4\sum_{\mathbf{x}_j \in \rho(\mathbf{e}_i)} \mathbf{x}_j\right\}. \tag{3.5}$$

3.4.3 Cell Rule

For each octahedral cell $\mathbf{o} = [\mathbf{x}_{i_0}, ..., \mathbf{x}_{i+5}]$ with the major diagonal $\mu(\mathbf{o}) = [\mathbf{x}_{j_1}, \mathbf{x}_{j_2}]$, the new cell point $\mathbf{c}_{new}$ is computed by:

$$\mathbf{c}_{new} = \frac{1}{32}\left\{4(\mathbf{x}_{i_0} + \cdots + \mathbf{x}_{i+5}) + 4(\mathbf{x}_{j_1} + \mathbf{x}_{j_2})\right\}. \tag{3.6}$$

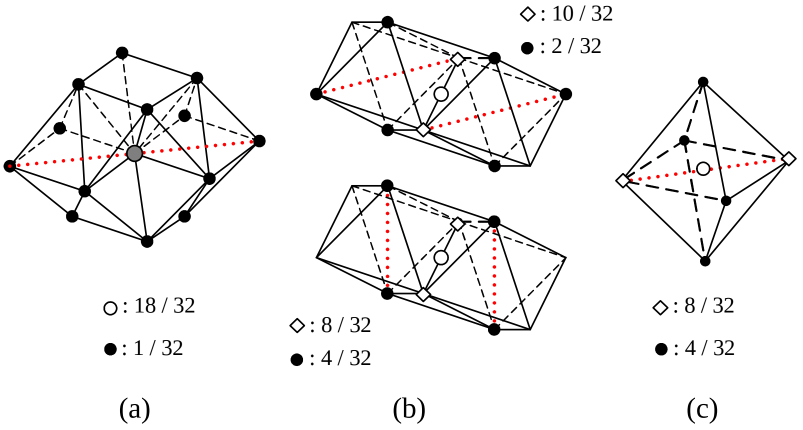


Fig. 12. The regular subdivision rules. The red dotted edges indicate the major diagonals.

3.5 Extraordinary Subdivision Rules

From the regular subdivision rules that are defined over the structured meshes based on the octet-truss, we extend our scheme to arbitrary tetrahedral meshes. Unlike the octet-truss, where each vertex and edge have a regular number of adjacent vertices, we can have arbitrary number of vertices adjacent to vertices and edges in the tetrahedral meshes. In solid subdivision, the extraordinary cases include two different types of irregularity, namely the extraordinary vertices and the extraordinary edges. However, once subdivided, the sub-structure inside each cell becomes regular again. Therefore, the extraordinary topology is limited to the initial connectivity.

The following generalization of the subdivision rules are simply based on weight averaging. We divide the subdivision coefficients for the 1-ring neighbor vertices by the valence. They require analysis to guarantee a certain level of continuity across the extraordinary topologies. The analysis for each case is described in Section 4.

3.5.1 Vertex Rule

Similar to subdivision surfaces, an arbitrary tetrahedral mesh can contain extraordinary vertices as shown in Figure 13(a). Suppose the valence of the vertex $\mathbf{x}$ is k ($|\rho(\mathbf{x}_i)| = k$), then our vertex rule can be rewritten as:

$$\mathbf{v}_{new} = \frac{9}{16}\mathbf{x}_i + \frac{7}{16k}\sum_{\mathbf{x}_j \in \rho(\mathbf{x}_i)} \mathbf{x}_j. \tag{3.7}$$

3.5.2 Edge Rules

The extraordinary edge case is not present in subdivision surfaces. Suppose the edge $\mathbf{e} = [\mathbf{x}_i, \mathbf{x}_{i+1}]$ is surrounded by k vertices ($|\rho(\mathbf{e}_i)| = k$), then the edge rule is modified as follows:

$$\mathbf{e}_{new} = \frac{5}{16}(\mathbf{x}_i + \mathbf{x}_{i+1}) + \frac{3}{8k}\sum_{\mathbf{x}_j \in \rho(\mathbf{e}_i)} \mathbf{x}_j. \tag{3.8}$$

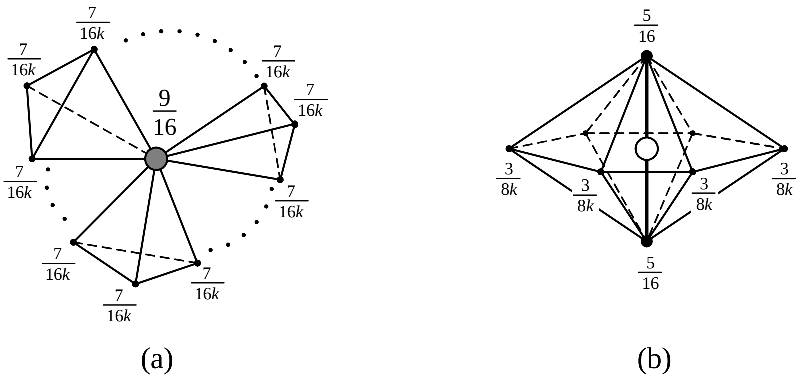


Fig. 13. The extraordinary subdivision rules.

Solid objects always have boundaries in 3D space. Therefore, we need to employ special rules to represent the boundaries of the solid objects represented by our subdivision algorithm. We simply use the modified Loop's scheme (Figure 14) to represent the boundaries. Because of the small sizes of our subdivision masks, there is no apparent problem between the transition area of the boundary and the interior. Since the Loop's scheme is based on the bivariate double directional box spline, the choice seems appropriate. It should be mentioned that the boundary representation is solely chosen by the visual quality. We have made no attempt to analyze the continuity between the boundary and interior of the solid objects.

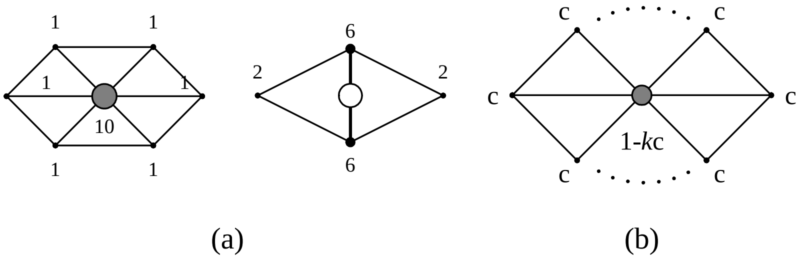
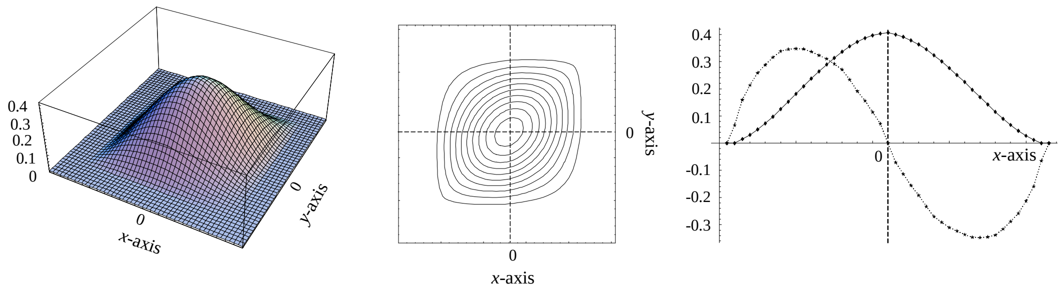


Fig. 14. The boundary subdivision rules based on the Loop's scheme. (a) The rules for the regular meshes. The values are to be multiplied by $\frac{1}{16}$. (b) The rule for extraordinary vertices. c can be chosen as $\frac{3}{8k}$ or $\frac{1}{k}(\frac{5}{8} - (\frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{k})^2)$.

4 Analysis of Arbitrary Topology

By definition, our scheme is C^2-continuous on the octet-truss structured meshes, since it evaluates the trivariate double-directional box spline functions defined on each regular vertex. Figure 15 shows a single basis function evaluated by our subdivision algorithm on the regular mesh. Figure 15(a) shows the density of the basis function cut by the x-y plane. Figure 15(b) is the iso-contour lines of the same function. Since we choose the direction of the line $y = x$ as our major diagonal direction, the shape is symmetry along the line. Figure 15(c) is the function value and the directional derivative $\frac{\partial f}{\partial x}$ along the x-axis. The plot data are from the 4th level of subdivision. The derivative is acquired by the central differences of the discrete data.

The convergence and the continuity across the extraordinary topology requires separate analysis. Many researchers have been working on the subdivision analysis near extraordinary topology, especially for the subdivision surfaces. For instance, Dyn and Micchelli (1990); Dyn et al. (1991, 1992), Micchelli and Prautzsch (1987), Prautzsch (1985); Prautzsch and Umlauf (1998); Prautzsch and Reif (1997); Prautzsch (1998); Prautzsch and Reif (1999), Reif (1995b,a), Zorin et al. (1996);



(a) (b) (c)

Fig. 15. The basis function of the subdivision algorithm on the regular mesh. (a) The density values of the basis function over the x-y plane. (b) The iso-contour lines of the density. (c) The function value and its directional derivative along the x-axis. The derivative values have been re-scaled.

Zorin (1997, 2000); Zorin and Kristjansson (2002), and most recently Levin and Levin (2003) investigated the sufficient and necessary conditions of the convergence and the C^1-continuity for various subdivision curves and surfaces. Two major techniques of such analysis are the spectral analysis of subdivision matrices and the characteristic map method by Reif (1995b). Unfortunately, these analysis techniques for subdivision surfaces are not fully extended to the solid subdivision algorithms and there is no known method specially developed for the solid schemes. Hence, we approach this situation as follows:

(a) Categorize the extraordinary cases for the solid subdivision algorithm.
(b) For each case, compute the subdivision matrix.
(c) Perform the spectral analysis of the subdivision matrix numerically.
(d) Construct the characteristic map and confirm the satisfactory conditions for the C^1-continuity through empirical data.

Even though these steps do not guarantee the continuity of the subdivision algorithm on every possible case, they suggest a strong evidence that our subdivision algorithm is indeed C^1-continuous for many situations, especially that can be occurred in real-world.

Since our subdivision algorithm is based only on stationary linear combinations, we can describe the subdivision process of each step as a simple matrix computation:

$$\mathbf{p}^{j+1} = \mathbf{S}\mathbf{p}^{j}, \tag{4.1}$$

where $\mathbf{p}^j = [\mathbf{p}^j{}_0, ..., \mathbf{p}^j{}_N]^T$ is a matrix of control points around $\mathbf{p}^j{}_0$ at the subdivision level j. The number N of the control points is determined so that the linear system is invariant. In our case, we need 2 rings of vertex neighbors.

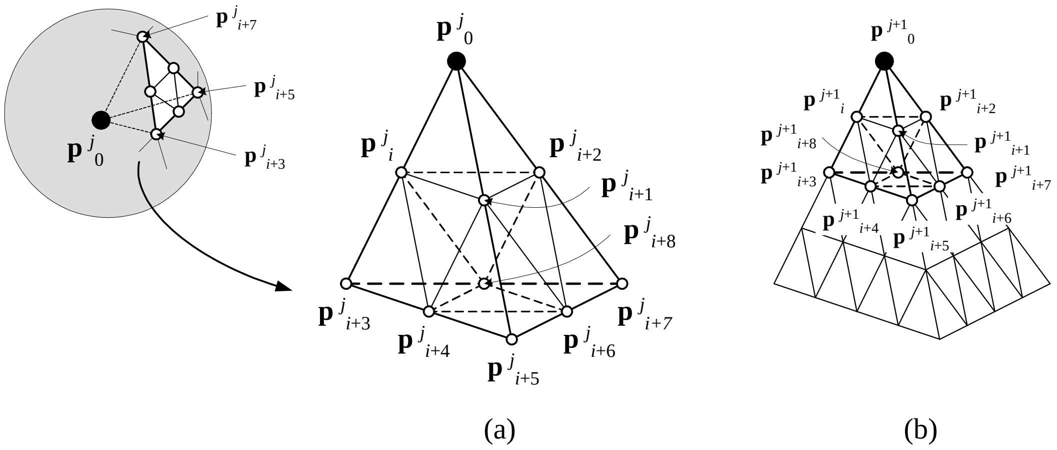


Fig. 16. The invariant neighborhood of an extraordinary vertex and their indices.

4.1 Subdivision Matrix

We begin the analysis with computing the subdivision matrix for each extraordinary case. We first examine the case of extraordinary vertices. This case involves a vertex with k vertices adjacent to it. As shown in Figure 16, we can establish a correspondence between the k adjacent vertices and k vertices on the sphere centered by the extraordinary vertex $\mathbf{p}^j{}_0$. By considering different triangulations of the k vertices, we can understand the different configurations of the extraordinary vertex subdivision matrix. Each triangle is associated with the tetrahedral area that is surrounding the extraordinary vertex. Because we need 2-ring vertex neighbors to acquire the invariant system, we subdivide each tetrahedron once, as illustrated in Figure 16. Using the Poincaré formula and the relation between triangular faces and edges:

$$\begin{aligned} v - e + f &= 2, \\ 2e &= 3f, \end{aligned}$$

we can deduce that the number of such tetrahedral area surrounding the vertex is $f = 2(k-2)$. In addition, the 1-ring vertex neighbor contains k vertices and each subdivided triangular faces on the 2-ring vertex neighbor contains 6 vertices, 3 of which are shared by each edge. Therefore, the actual number N of the vertices including the extraordinary vertex to form the invariant system is:

$$\begin{aligned} N &= 1 + k + 6f - 3e + k \\ &= 1 + k + 6(2k-4) - 3(3k-6) + k = 5k - 5. \end{aligned}$$

Hence, we can conclude that the size of the subdivision matrix for each extraordinary vertex with the valence k is $N \times N$ where $N = 5k - 5$. With a proper reordering of the indexes of the vertices, the matrix $\mathbf{S}_v$ can be written as:

$$\mathbf{S}_v = \begin{pmatrix} \mathbf{M} & \mathbf{O} \\ \mathbf{A} & \mathbf{B} \end{pmatrix},$$

where $\mathbf{M}$ is a $(k+1) \times (k+1)$ matrix and $\mathbf{O}$ is the square zero matrix with the size of $4k-6$. We use the spherical coordinates to define an order between vertices. It is important to know that the dominant and the subdominant eigenvalues of the $\mathbf{S}_v$, especially the first 5 largest eigenvalues, are identical to those of the submatrix $\mathbf{M}$. Since the matrix $\mathbf{M}$ can be easily acquired by the k 1-ring vertex neighbors of the vertex ${\mathbf{p}^j}_0$, we can reduce the amounts of the computations during the analysis process significantly. It is worth mentioning that, unlike the surface cases, there exist several different configurations of neighboring vertices for each valence k. In fact, it is related to the planar triangulation of k points and the recent result by Santos and Seidel (2003) suggests that the upper bound for the number of the configurations is $O(59^n n^{-6})$ for large n. Since each configuration yields a unique subdivision matrix, it is difficult to compute the eigensystem systematically.

The extraordinary edge with the valence k is surrounded by k tetrahedra sharing the edge $e = [{\mathbf{p}^j}_0, {\mathbf{p}^j}_2]$, as shown in Figure 17. Again, we subdivide each tetrahedron once to make the neighbor invariant. It is easy to deduce that the size of the subdivision matrix $\mathbf{S}_e$ is $(4k+3) \times (4k+3)$. Similar to the extraordinary vertex subdivision matrix, the matrix $\mathbf{S}_e$ can be described as:

$$\mathbf{S}_e = \begin{pmatrix} \mathbf{L} & \mathbf{O} \\ \mathbf{P} & \mathbf{Q} \end{pmatrix},$$

with the proper index reordering. In the edge case, $\mathbf{L}$ is a $(2k+3)\times(2k+3)$ matrix. It consists of the subdivision coefficients of the 1-ring neighbors of the extraordinary edge. Once more, the dominant and subdominant eigenvalues of the subdivision matrix $\mathbf{S}_e$ can be acquired from the submatrix $\mathbf{L}$. The subdivision matrix $\mathbf{S}_e$ and its eigensystem can differ by the choice of the major diagonals. It will be discussed in the next section.

4.2 Prerequisites

There is a question to be answered before continuing the spectral analysis. It is the question about the face-to-face case between the tetrahedral cells. Even though a tetrahedron always faces with an octahedron and vice versa in the regular octet-truss meshes, this property does not hold in general. Especially, an arbitrary tetrahedral mesh does not satisfy it at all. Once subdivided, the interior of each tetrahedron becomes the regular octet-truss structure. However, the faces shared by two tetrahedra initially given by the mesh remain the same during the subdivision processes.

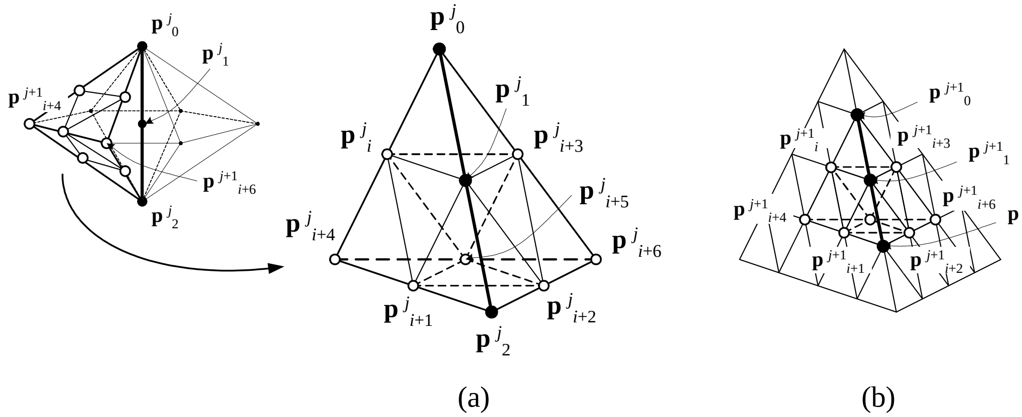


Fig. 17. The invariant neighborhood of an extraordinary edge and their indices.

In fact, in the paper (Schaefer et al., 2004), Schaefer et al. resolved the issue by applying the joint spectral radius test by Levin and Levin (2003). Our situation is slightly different, since their rules are based on the cell-averaging, while ours are based on the vertex and edge-averaging. Moreover, the choice of the major diagonals plays a major role to keep the structures in the shared face regular.

We have taken a simpler approach to guarantee the continuity of the face-to-face case. We argue that, with the proper choice of the major diagonals as defined in Definition 3.2, the vertices and edges on the shared face admit the subdivision matrices of the regular cases. We begin with two tetrahedra as shown in Figure 18(a). To acquire the invariant neighbors for each vertex in the shared face, we subdivide the tetrahedra 3 times. The 1-ring vertex neighbor of each yellow vertex in Figure 18(b) consists of 6 adjacent vertices on the face and 2 adjacent vertices in each facing tetrahedron. In addition, we can have 2 more adjacent vertices, decided by the choice of the major diagonals. Therefore, each vertex has 14 neighbors in total which form a regular vertex case. For the shared edges, each edge on the shared face is surrounded by two adjoining octahedra and two adjoining tetrahedra. If we choose the major diagonals of the two octahedra *properly*, we can prove that each edge has the correct number of neighbors, either 6 or 4, with respect to its relative position against both major diagonals. All the possible cases are illustrated in Figure 19. In Figure 19(a) and (b), the edge has 6 neighbors, whereas it has 4 neighbors in Figure 19(c).

We support our argument with some empirical results. Figure 20 offers a visual confirmation on the continuity between the tetrahedra. We evaluate the basis function centered on one of the shared vertices in between. The shared face of the two tetrahedra are located on the x-y plane. One tetrahedron is placed on the negative z side, while the other is placed on the positive side. The major diagonals of the tetrahedra are chosen so that it is in the proper situation defined in Definition 3.2. The values are evaluated from the 3 levels of the subdivision algorithm. Figure 20 shows very smooth transition between one tetrahedron to another.

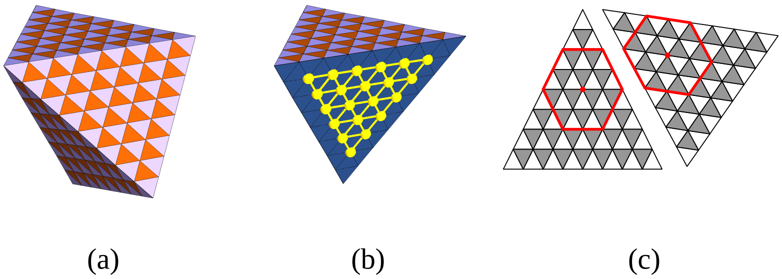


Fig. 18. An example of the face-to-face case. (a) Two tetrahedra share the face. The orange-colored faces indicate the faces from octahedral cells. (b) The vertices in between. The yellow vertices has 14 neighbors with the correct choice of the major diagonals. (c) The 2-ring neighbor of the shared vertex.

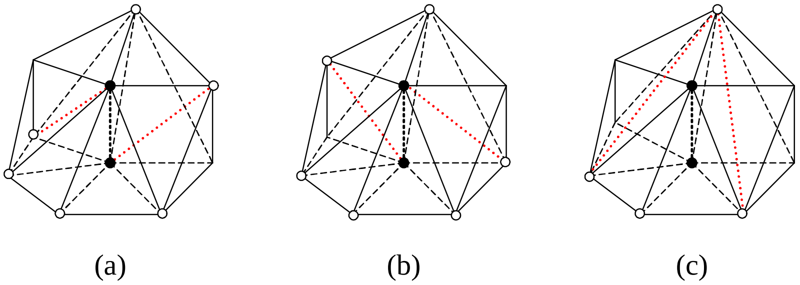


Fig. 19. The different neighbors of the edge between faces by the proper choices of the major diagonals. The red dotted lines are the major diagonals.

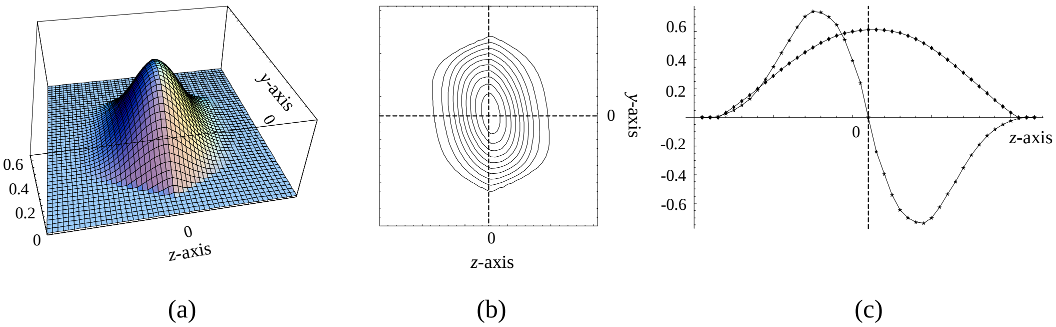


Fig. 20. The evaluation of the face-to-face case. (a) The density values of the basis function centered on the shared vertex over the y-z plane. (b) The iso-contour lines of the density. (c) The function value and its directional derivative along the z-axis. The derivative values are re-scaled.

The face-to-face case entails the question of the choice of the major diagonals in arbitrary meshes. The above results suggest that the proper choice of the major diagonals are important for the continuity of the face-to-face case. It is possible that the choice cannot satisfy the properties in Definition 3.2 globally for certain meshes. Thus, we should implement the algorithm carefully so that the choice is as proper as possible at the initial stage. We discuss it in Section 5.

Suppose the matrix $\mathbf{S}$ has the eigenvalues $\lambda_0 \geq \cdots \geq \lambda_l$ in non-increasing order with the associated eigenvectors $\mathbf{v}_0, ..., \mathbf{v}_l$. The matrix $\mathbf{S}$ is $N \times N$ matrix and each eigenvectors are in $\mathbb{R}^N$. Now, the original control points $\mathbf{p}^0$ can be rewritten in the eigenspace:

$$\mathbf{p}^0 = \sum_i \mathbf{a}_i \mathbf{v}_i, \tag{4.2}$$

and therefore, any level j vertices $\mathbf{p}^j$ can be represented by:

$$\mathbf{p}^j = \mathbf{S}^j \mathbf{p}^0 = \sum_i (\lambda_i)^j \mathbf{a}_i \mathbf{v}_i, \tag{4.3}$$

with each coordinate coefficient matrix $\mathbf{a}_i$. The coefficient $\mathbf{a}_i$ can be computed by $\mathbf{a}_i = \mathbf{v}_i \cdot \mathbf{p}^0$. The conditions of the eigenvalues and eigenvectors for the subdivision matrix are well understood and it is independent for each coordinate operation. Therefore, we hypothesize that the similar conditions still hold for our solid scheme. The conditions are as follows:

(a) λ_0 should be equal to 1 for the subdivision to be invariant with respect to translations and rotations.

(b) λ_i should be strictly less than 1 when $j > 0$ for the convergence of the scheme. The limit positions for the original control points is then

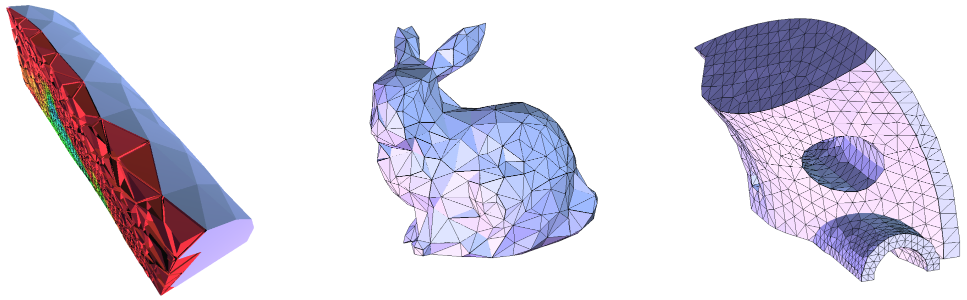
$$\mathbf{p}^0 = \lim_{j \to \infty} \mathbf{p}^j = \sum_i \left(\lim_j (\lambda_i)^j \mathbf{a}_i \mathbf{v}_i \right) = \mathbf{a}_0 \mathbf{v}_0. \tag{4.4}$$

(c) λ_4 should be strictly less than λ_3. Suppose the subdominant eigenvalues $\lambda_1 = \lambda_2 = \lambda_3 = \lambda > \lambda_4$ and the vertex $\mathbf{p}^0{}_0$ is the origin. Then we get:

$$\frac{\mathbf{p}^j}{\lambda^j} = \mathbf{a}_1 \mathbf{v}_1 + \mathbf{a}_2 \mathbf{v}_2 + \mathbf{a}_3 \mathbf{v}_3 + \mathbf{a}_4 \left(\frac{\lambda_4}{\lambda} \right)^j + \cdots , \tag{4.5}$$

which means the control points are approaching a fixed configuration up to a scaling factor λ^j. The remainders converge to zero in the limit since $\lambda > \lambda_j$ for $j \geq 4$. From this observation, we can derive that the three subdominant eigenvalues determine the behavior of the derivatives at $\mathbf{p}^0{}_0$ in the limit. This is a sufficient condition to define the characteristic map.

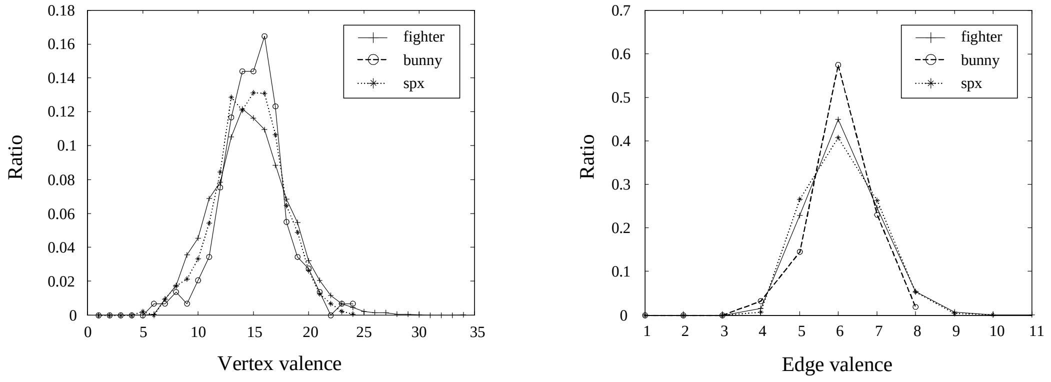
To verify these conditions, eigenvalues and eigenvectors of each subdivision matrix should be computed. As first observed in (Doo and Sabin, 1978), the subdivision matrix for an extraordinary vertex in surfaces cases has a cyclic structure due to its planar symmetry. Therefore, we can apply discrete Fourier transform to compute the eigen-structure of every valence systematically. Unfortunately, this is no longer true for the solid cases, as discussed in the previous section. Hence, we need to rely on numerical computation of the eigen-structure for specific cases. Since we cannot compute all the possible cases, we begin with statistics on vertex and edge valences of arbitrary tetrahedral meshes to determine which valence numbers should be included in our analysis. We choose few existing tetrahedral models (Figure 21) to examine the valence numbers for each vertex and edge. The meshes are acquired by various tessellation methods including the 3D Delaunay triangulation and the advance front technique.



(a) (b) (c)

Fig. 21. A selection of arbitrary tetrahedral meshes. (a) A cross-section of a fighter model with its environment (fighter). (b) Tesselated Stanford bunny (bunny). (c) A model of a machenical part (spx)

As Figure 22 Table 21 show, the distribution of the valence numbers are concentrated on the regular valence numbers. Moreover, the averages are very close to the regular valence numbers and the deviations are relatively small. These results suggest that in real world application, we only need to analyze relatively small number of valence cases. In this paper, we choose to analyze the valence number from 5 to 22 for the extraordinary vertices and the valence number from 4 to 9 for the extraordinary edges.



(a) (b)

Fig. 22. Distribution charts of the valence numbers of the selected arbitrary meshes. (a) Vertex valence. (b) Edge valence.

Table 2 presents the list of the first 6 eigenvalues of the selected extraordinary vertex cases. As we have mentioned, there exist several different configurations for each valence. We only show few cases in the table. Table 3 shows the list of the first 6 eigenvalues of the selected extraordinary edge cases. Unlike the vertex cases, each

Model name	No. vertices	No. edges	No. cells	V−val average	V−val maximum	E−val average	E−val maximum
fighter	13832	87587	70125	13.75	33	5.12	10
bunny	575	2904	1903	13.95	23	5.06	7
spx	2896	17212	12936	13.65	23	5.09	9

Table 1
Statistics on the valence numbers of the selected arbitrary meshes

valence has unique configuration. However, it differs by the choice of the major diagonals. Figure 23 shows two different choices of the major diagonals for the extraordinary edge with the valence 9. In the first row, the diagonals are chosen such that they are in skew positions mutually. In the second row, the diagonals point toward the center of the extraordinary edge. As Figure 23(b) and (c) suggest, the first choice yields much smoother results. The eigenvalues of the first case are shown in Table 3, whereas the 6 dominant eigenvalues of the second case are:

$$\{1., 0.619939, 0.619939, 0.528431, 0.528431, 0.5\}.$$

The difference between two cases becomes more apparent as the valence is getting higher. In fact, the characteristic map of the second case fails to confirm C^1 continuity for the valence larger than 8, as discussed in the next section. The values in Table 3 are taken from the configurations similar to the skew case for each valence. During the implementation of the subdivision algorithm, we try to optimize the choice of the major diagonals so that it produces the similar result as the first row for the most of the extraordinary edges. In Appendix B.1 and B.2, we present the subdivision matrices and their eigenvalues for the lowest valence cases in the tables. Two different choices of the major diagonals and their subdivision matrices are shown in Appendix B.2.

It is clear from the computed values that the eigenvalues of the most of the extraordinary cases satisfy the eigenvalues conditions for subdivision algorithms. However, the results so far only guarantee the convergence of the subdivision algorithm at the particular vertex or edge. In the next step, we examine the characteristic map of each case to verify C^1 continuity near the extraordinary vertex or edge.

4.4 Characteristic Map

If the eigenvalues satisfy $\lambda_0 = 1 \gneqq \lambda_1 \geq \lambda_2 \geq \lambda_3 \gneqq \lambda_4$, we are able to apply the characteristic map method by Reif (1995b) to prove C^1 continuity. From the real eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ associated with the first 3 subdominant eigenvalues, we define a map:

$$\Psi = N[\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3] : U \times \mathbb{Z}_k \longrightarrow \mathbb{R}^3, \tag{4.6}$$

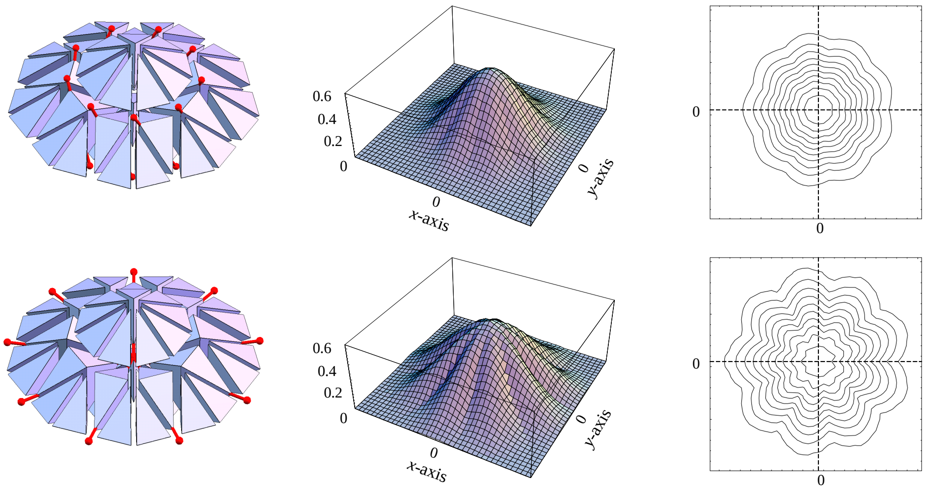
Valence	λ_0	λ_1	λ_2	λ_3	λ_4	λ_5
5	1.	0.3125	0.292083	0.15	0.125	0.125
6	1.	0.312499	0.25	0.25	0.25	0.15
7	1.	0.327254	0.327254	0.3125	0.275888	0.15
8(a)	1.	0.480205	0.3125	0.3125	0.249998	0.2375
8(b)	1.	0.375	0.375	0.3125	0.270178	0.15
9	1.	0.405872	0.405872	0.3125	0.26545	0.19437
10(a)	1.	0.477404	0.418566	0.418566	0.2375	0.206434
10(b)	1.	0.426777	0.426777	0.3125	0.261451	0.25
11	1.	0.441511	0.441511	0.3125	0.293412	0.293412
12	1.	0.480205	0.480205	0.480205	0.250002	0.2375
13	1.	0.460313	0.460313	0.353854	0.353854	0.3125
14(a)	1.	0.577132	0.449431	0.449431	0.34832	0.3125
14(b)	1.	0.517404	0.517404	0.480205	0.3125	0.3125
15	1.	0.471364	0.471364	0.392016	0.392016	0.3125
16	1.	0.541169	0.541169	0.480204	0.372645	0.372645
17	1.	0.571212	0.511703	0.511703	0.371472	0.358853
18(a)	1.	0.623289	0.463128	0.463128	0.457191	0.374739
18(b)	1.	0.557148	0.557148	0.480205	0.418566	0.418566
20(a)	1.	0.571212	0.549072	0.549072	0.3875	0.3875
20(b)	1.	0.568361	0.568361	0.480206	0.453454	0.453454
22	1.	0.616629	0.525774	0.525774	0.4625	0.427853
22(b)	1.	0.576511	0.576511	0.480205	0.480205	0.480205

Table 2
Eigenvalues for a selection of the extraordinary vertex cases.

Valence	λ_0	λ_1	λ_2	λ_3	λ_4	λ_5
4	1.	0.477404	0.418566	0.418566	0.2375	0.206434
5	1.	0.480205	0.480205	0.480205	0.25	0.2375
6	1.	0.517404	0.517404	0.480205	0.3125	0.3125
7	1.	0.541169	0.541169	0.480204	0.372645	0.372645
8	1.	0.557148	0.557148	0.480205	0.418566	0.418566
9	1.	0.568361	0.568361	0.480206	0.453454	0.453454

Table 3
Eigenvalues for a selection of the extraordinary edge cases.

Where U is a unit simplex in 3D pivoted on the origin with the corner containing the origin has been removed at the half points of the edges. Figure 24 illustrates an example of the control net of the characteristic map and its cross-section. Note that the characteristic map by Reif et al.is originally defined on 2D space. We generalize the map to the solid case and assume that it is correct without any proof. Now, we



(a) (b) (c)

Fig. 23. Two different choices of the major diagonals for the extraordinary edge with the valence 9. (a) The different choices of the major diagonals indicated in the red lines. (b) The density values. (c) The Iso-contour lines.

presume that if we can prove that the characteristic map Ψ is regular and injective, then our subdivision algorithm satisfies C^1 continuous. For 2D, the regularity and injectivity can be proven by considering the complex plane. However, due to lack of any planar symmetry in general, the direct analogy of the method has been proven to be difficult in 3D. Instead, we rely on the experimental results to conform its regularity and injectivity. For the selected cases, we perform the subdivision and prolongation (Reif, 1995b) successively, up to certain levels, followed by the visual inspection of the results. In most cases, it has been relatively easy to deduce that the process will not produce any irregularity or self-intersections. In contrast, Figure 27 shows the characteristic maps for the valence 9 and 11 cases with specific choice of the major diagonals. In these cases, the diagonals are chosen to point toward the center, as shown in the second case of Figure 23. During the subdivision and prolongation process, the maps form a single saddle surface and therefore they are not injective. In these two cases, we cannot determine the C^1 continuous of the subdivision algorithm near the extraordinary edges by the characteristic map method.

(a) (b)

Fig. 24. The control net of the characteristic map of the extraordinary vertex with the valence 11. (a) The control net. (b) The cross-section of the control net.

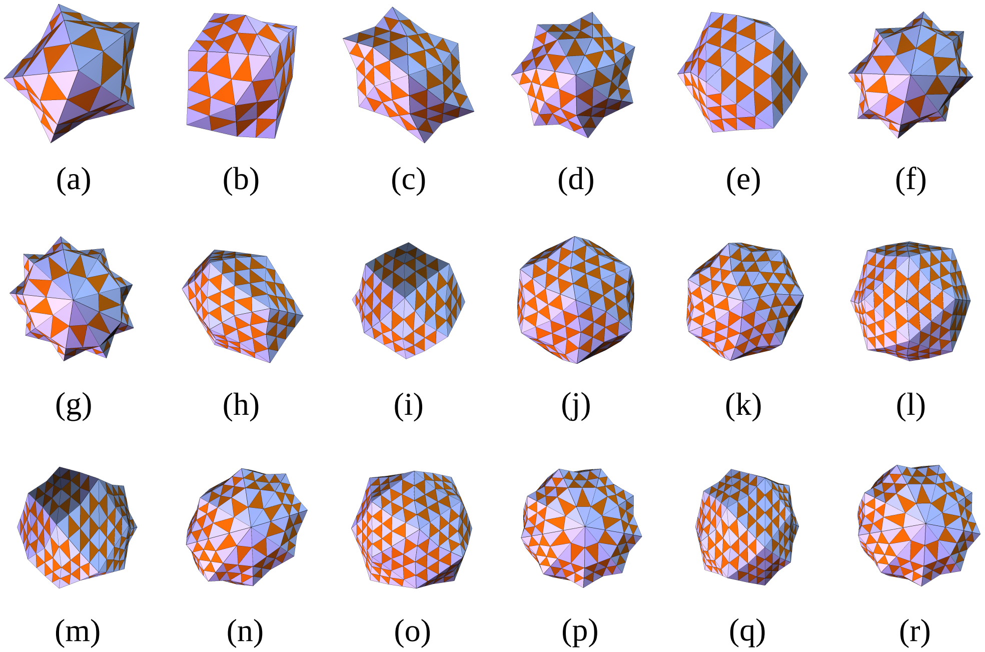


Fig. 25. Control nets for a selection of the characteristic maps of the extraordinary vertex with the valences from 7 to 22. The orange-colored faces indicate the faces from octahedral cells.

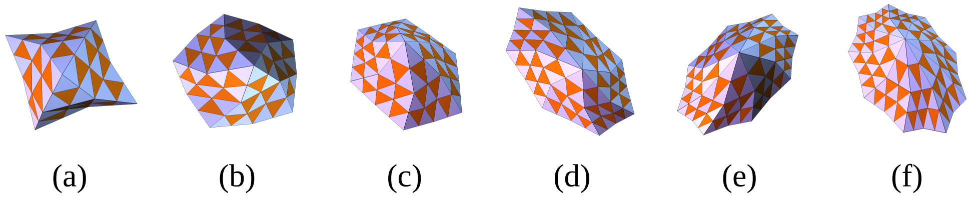


Fig. 26. Control nets for a selection of the characteristic maps of the extraordinary edges with the valences from 4 to 9. The orange-colored faces indicate the faces from octahedral cells.

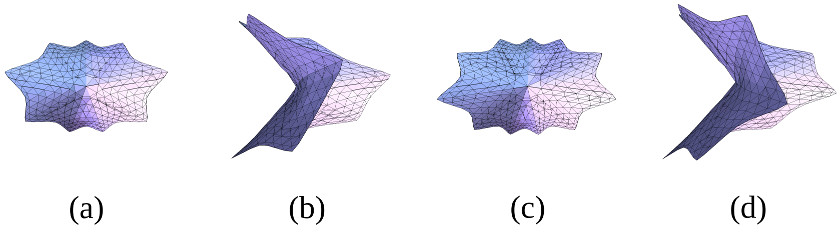


Fig. 27. The characteristic maps for the extrarodinary edges with the valences 9 and 11. The major diagonals are chosen to point toward the center. The maps are not injective. (a-b) The valence 9 case. (c-d) The valence 11 case.

```
Octet-Subdivision ({$v_i$}, {$e_j$}, {$c_k$})
{Input {$v_i$}, {$e_j$}, {$c_k$}: A set of vertices, edges, and cells, respectively}
if this is the first subdivision then
    call Major-Diagonal-Choose ({$c_k$})
end if
for all vertex $v_i$ do
    compute a new vertex point $p_i$
end for
for all edge $e_j$ do
    compute a new edge point $q_j$
end for
for all cell $c_k$ do
    if $c_k$ is tetrahedral then
        split $c_k$ into 5 subcells $c^0{}_k,\ ..., c^4{}_k$
    else if $c_k$ is octahedral then
        compute a new cell point $r_k$
        split $c_k$ into 14 subcells $c^0{}_k,\ ..., c^{13}{}_k$
    end if
    inherit the major diagonal choice from $c_k$
    construct new edges {$e'_l$} from $\{p_i\} \cup \{q_j\} \cup \{r_k\}$
end for
return $\{p_i\} \cup \{q_j\} \cup \{r_k\}$, {$e'_l$}, {$c^m{}_k$}
```

Algorithm 1. Octet-Subdivision.

5 Implementation

The implementation of the subdivision algorithm is straightforward, except the maintenance of the major diagonals. As outlined in Algorithm 1, the subdivision object is represented by the vertices, edges and cells. The edges provide the connectivity information between the vertices. The major diagonal information is stored with each cell. For tetrahedral cells, the major diagonal is represented by a pair of non-adjacent edges. For octahedral cells, the major diagonal is represented by a pair of non-adjacent vertices. Since there are only 3 choices for both cases, only 2 bits of additional memory is required for each cell. For each vertex and edge, we compute new vertex and edge point by the subdivision rules, using their neighbors. Then, we split each cell into subcells using the new vertices. During the split, if the cell is octahedral, we need to compute the cell point. Each subcell inherits the information on the major diagonal from its parent cell.

If the subdivision is performed for the first time with an arbitrary tetrahedral mesh, there is no given major diagonal information. In this case, we use a function Major-Diagonal-Choose (Algorithm 2) to choose the major diagonal for each cell properly. The algorithm is based on the breath-first search of the adjacency tree of

```
MAJOR-DIAGONAL-CHOOSE ($\{c_i\}$)
{Input $\{c_i\}$: A set of tetrahedral cells
   $c_i$.visited; true if the cell is visited during the search.
   $c_i$.edges; A set of edges of the cell.
   $c_i$.major-diag; A major diagonal of the cell
   ($c_i$.major-diag $\subset$ $c_i$.edges, $|c_i$.major-diag$| = 2$). }
initialize a queue $q$ and a set $s$
$q$.push($c_0$)
while $q \neq \varnothing$ do
   $c \leftarrow q$.pop()
   $s \leftarrow \varnothing$
   for each adjacent cell $c'$ of $c$ do
      if $c'$.visited is true then
         $s \leftarrow s \cup c'$.major-diag
      else
         $q$.push($c'$)
      end if
   end for
   if $|c$.edges$-s| > 1$ then
      choose $c$.major-diag from $c$.edges$-s$
   else
      choose $c$.major-diag arbitrary
   end if
   $c$.visited $\leftarrow$ true
end while
return
```

Algorithm 2. MAJOR-DIAGONAL-CHOOSE.

the cells. During the search, it tries to minimize the conflict of the major diagonal choice between adjacent cells, so that the choice is as proper as possible globally.

5.1 Experimental Results

A major benefit of our solid subdivision scheme is the ability to represent complex solid models with heterogeneous materials. Figure 28 shows a selection of subdivision models designed by a simple modeling tool that we developed for the subdivision scheme. Models with complex topology can be easily represented with the unified scheme. Not only the boundaries, but also the internal structures are smooth and well-defined. Figure 28(f) illustrates a model with non-trivial topology, which cannot be represented by surface subdivision schemes without serious modification. Our algorithm can handle it with no exceptional rule, since we make no special assumption on the connectivity of 3D meshes.

Figure 29 shows a torus model with hybrid dimensionality. It consists of solid and surface parts. With some additional rules, our scheme can be used for non-manifold object representations. Extensive research on non-manifold representation using the solid subdivision scheme is discussed in (Chang and Qin, 2004).

Figure 30 and Figure 31 present volumetric models with heterogeneous material properties associated with geometry. In Figure 30, the cylinder model has different material densities at the top and the bottom of the object. By utilizing our algorithm, we can blend the different densities smoothly. As Figure 30(c) shows, the result is much smoother than simple linear interpolation. In Figure 31, we assign tensions at each vertex on the coarsest level using simple Laplace's equation with initial condition. Each time step, instead of solving the equation on the fine levels, we simply apply our subdivision rules to interpolate the values using the coarsest level as initial values. The results are shown in Figure 31(b) and (c). Even though the potential has not been fully investigated, we believe that our subdivision scheme can serve as better blending functions or prolongation operators for finite element analysis or multigrid method based on tetrahedral meshes, instead of trivariate linear interpolation. It should be addressed in our future publication.

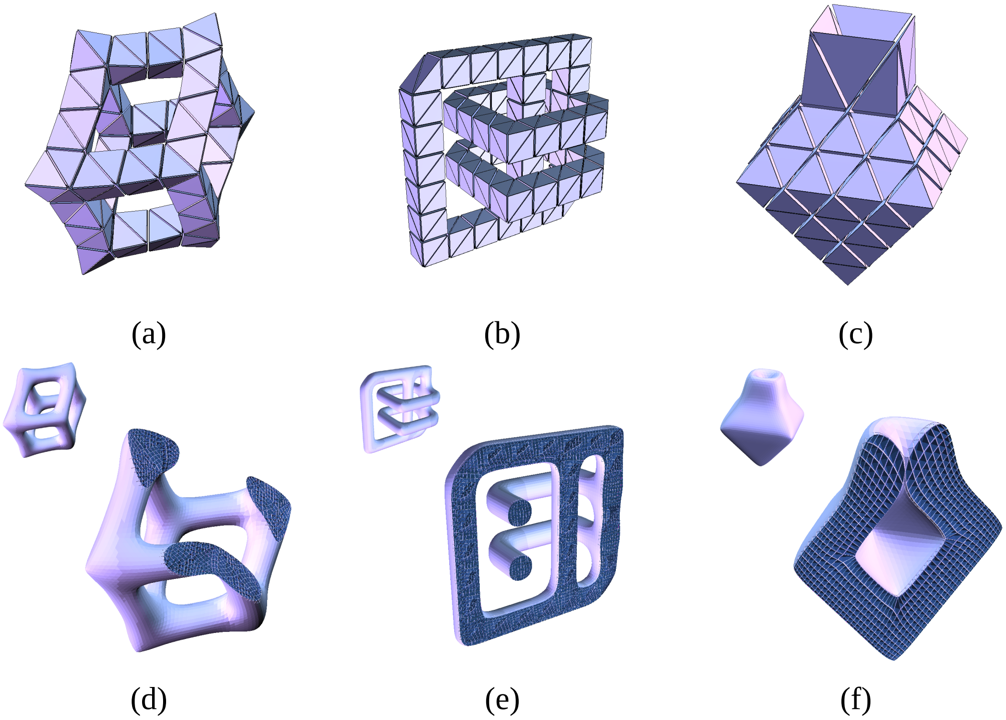


Fig. 28. Solid subdivision models with non-trivial topology. (a-c) Initial control meshes. (d-f) The models at subdivision level 3 and their cross-sections.

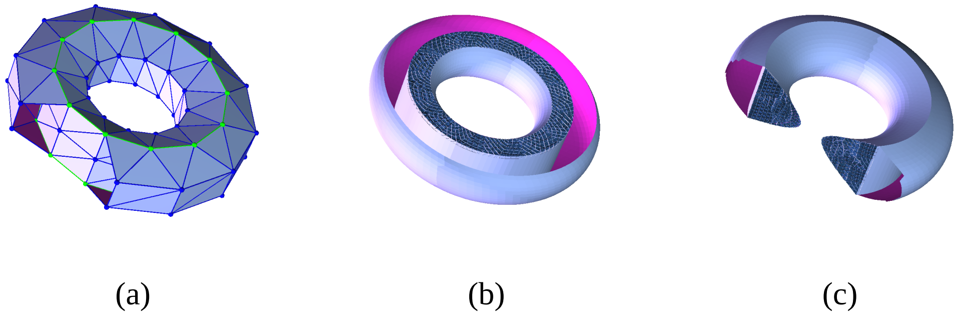


Fig. 29. A torus model that consists of a solid and a surface. (a) Initial control mesh. The cut shows the internal structure. Purple areas are the backside of a surface. (b) A cross-section of the model at subdivision level 3. (c) Another cross-section of the model at subdivision level 3.

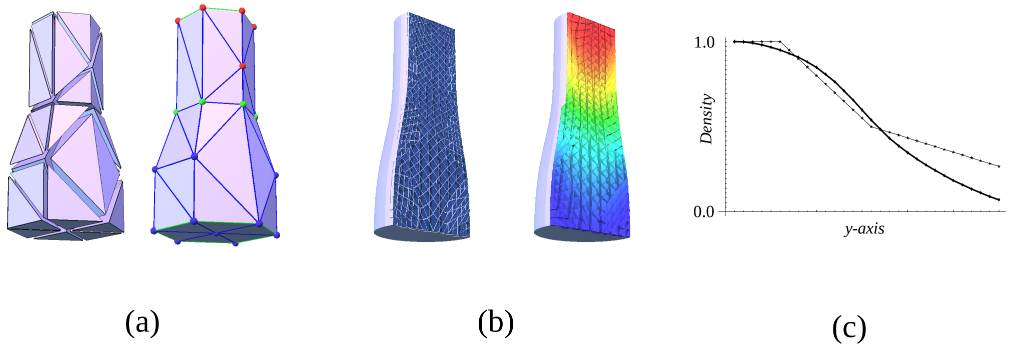


Fig. 30. A cylinderical model with heterogeneous material. (a) Initial control mesh and assigned material density (color-coded). (b) A cross-section of the model at subdivision level 3 and its density distribution. (c) Comparison of density distribution by our subdivision algorithm (bold line) and tri-linear interpolation (thin line).

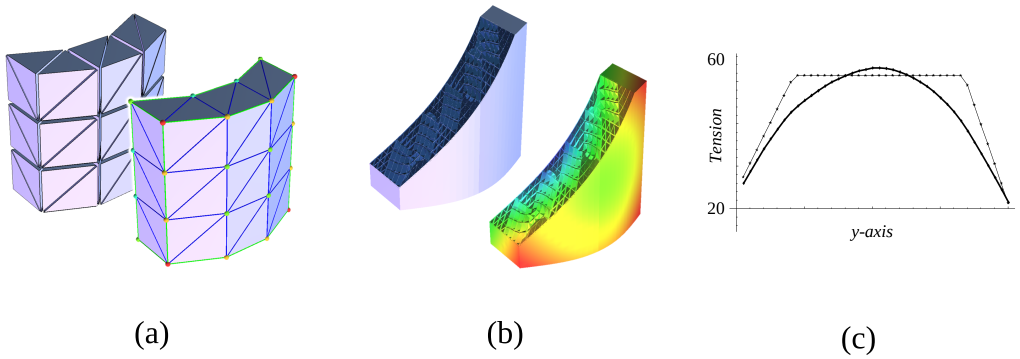


Fig. 31. A panel model with simulated tension force. (a) Initial control mesh and assigned tension (color-coded). (b) A cross-section of the model at subdivision level 3 and its tension interpolation. (c) Comparison of tension interpolation by our subdivision algorithm (bold line) and tri-linear interpolation (thin line).

6 Conclusion

We have developed novel solid subdivision schemes based on box splines over arbitrary tetrahedral meshes. In addition, their analysis, implementation issues, and experimental examples are presented. Trivariate box spline basis functions provide our scheme desirable continuity as well as a sound mathematical foundation. Defined over tetrahedral meshes, the proposed scheme is flexible, robust and easy to generalize. We have proved the convergence and continuity of the scheme by utilizing existing mathematical techniques as well as supportive empirical results. We have presented several modeling examples that our scheme can manage easily in comparison with other contemporary solid representations. These examples have been particulary chosen to address the specific issues in solid modeling provoked by the advances in engineering technologies. In particular, the advantages of the proposed schemes over existing representations are as follows: (a) Unified solid and boundary representation. (b) Multiresolution approach. (c) Free from topological restriction. (d) Ability to represent heterogeneous materials.

There are several problems that we have encountered during the research of the solid subdivision. First, unlike tensor-product subdivision schemes, a generalization of the scheme over higher dimensions is not always trivial. In fact, the general simplicial mesh subdivision is non-trivial issue for the dimension over 4. Unless we introduce specific types of polyhedral cells, it is not always possible to devise high dimensional subdivision schemes over simplicial meshes. Secondly, the continuity analysis requires better mathematical methods. Unlike the surface cases, general solid and high dimensional subdivision scheme analysis has not been researched intensively due to several difficulties and lack of necessity until now. In consequence, most of the related researches employ the similar approaches to ours; the existing techniques for the surface cases, such as spectral analysis and characteristic map method, and empirical results to support the continuity. None of these approaches are fully satisfactory for the further development of solid subdivision schemes. Therefore, new mathematical analysis techniques should be followed to guarantee the development of high dimensional subdivision schemes. Thirdly, the increase of the number of cells during the subdivision process always concerns the practical application of the subdivision scheme. In fact, the question of how to reduce the speed of the cell increase for 3 and higher dimensional space is far from trivial. There have been few suggestions, for instance (Pascucci, 2002), but not without their own problems. We are currently working the better refinement techniques to reduce the cell growth. Finally, there are few issues that we have not addressed in this paper, such as volumetric compression and fairing using our subdivision scheme. These applications are a part of our ongoing research.

A Averaged Subdivision Scheme

As proven in the previous sections, the choice of the major diagonal plays major role over both the regular structured meshes and the analysis over arbitrary tetrahedral meshes. However, there are some disadvantage involving the diagonals. First, for an arbitrary mesh, MAJOR-DIAGONAL-CHOOSE (Algorithm 2) does not return proper choice of the major diagonals *globally,* in general. Secondly, it requires meticulous bookkeeping on the entire meshes and across the subdivision levels. Finally, it generates some "favored" direction in the represented object, which can cause asymmetry. It is obvious because the basis function, or the box spline, produced by the subdivision algorithm on the structured mesh is *not* radially symmetric. This fact could cause some problem during heterogeneous material modeling, if the mesh and the diagonals are not carefully chosen.

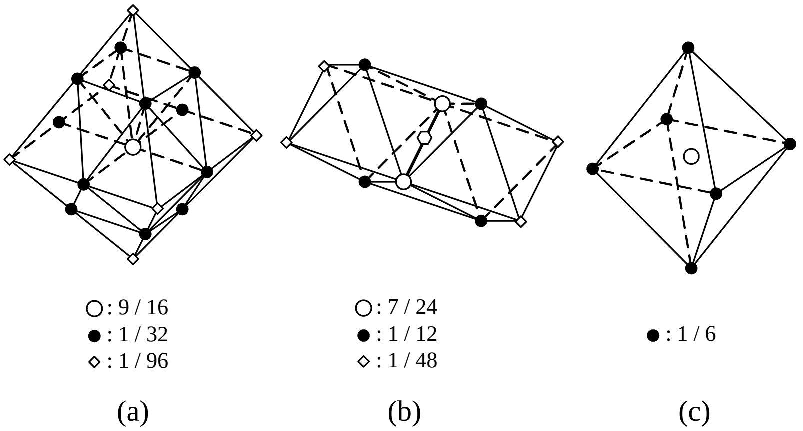
In fact, there is a simple solution to avoid the major diagonals by averaging the basis function of each direction. In $\mathbb{Z}^m$ with the octet-truss structure, there are 3 choices in the directions of the major diagonals, *i.e.*$(1,1,0)$, $(-1,1,0)$, and $(0,0,1)$. Suppose we describe our object S_Ξ associated the particular major diagonal direction defined by the Ξ as:

$$S_\Xi(\mathbf{x}) = \sum_{\mathbf{i}\in\mathbb{Z}^m} \mathbf{p}_\mathbf{i} M_\Xi(\mathbf{x}-\mathbf{i}), \tag{A.1}$$

where $\mathbf{p}_\mathbf{i}$ are initial control points. Let us denote the relevant directional matrices as Ξ_1, Ξ_2, and Ξ_3, respectively. Then, we can define the new object S without any major diagonals:

$$S(\mathbf{x}) = \frac{1}{3}\sum_{j=1}^{3} S_{\Xi_j}(\mathbf{x}). \tag{A.2}$$

It is easy to find the corresponding subdivision rules. One can simply alternate the major diagonals in the regular rules (Figure 12), sum the weights up at each vertex and divide it by 3. The computed values are shown in Figure A.1. We call the scheme the *averaged solid subdivision scheme*. Note that the face-to-face case becomes irregular with these averaged masks and the argument in Section 4.2 becomes invalid. The extraordinary analysis of this particular scheme requires a new mathematical tool, which has not been exploited yet.



(a) (b) (c)

Fig. A.1. The regular rules for the averaged subdivision scheme.

B Subdivision Matrices

B.1 Extraordinary Vertex Subdivision Matrix

The matrix $\mathbf{S}_v$ below is the subdivision matrix for an extraordinary vertex with the valence 5. The size of the matrix is 20×20. The subdivision matrices for extraordinary vertices with different valences have the similar structure.

$$\mathbf{S}_v = \left(\begin{array}{cccccccccccccccccccc}
\frac{9}{16} & \frac{7}{80} & \frac{7}{80} & \frac{7}{80} & \frac{7}{80} & \frac{7}{80} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{5}{16} & \frac{5}{16} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{4} & \frac{1}{8} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{4} & \frac{1}{8} & \frac{1}{8} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{4} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{4} & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{5}{16} & 0 & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{5}{16} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{7}{128} & \frac{9}{16} & \frac{7}{128} & \frac{7}{128} & \frac{7}{128} & 0 & \frac{7}{128} & \frac{7}{128} & \frac{7}{128} & \frac{7}{128} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{16} & \frac{5}{16} & \frac{5}{16} & \frac{1}{16} & \frac{1}{16} & 0 & 0 & \frac{1}{16} & 0 & 0 & \frac{1}{16} & 0 & \frac{1}{16} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{16} & \frac{5}{16} & \frac{1}{16} & \frac{5}{16} & \frac{1}{16} & 0 & 0 & 0 & \frac{1}{16} & 0 & 0 & 0 & \frac{1}{16} & 0 & \frac{1}{16} & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{16} & \frac{5}{16} & \frac{1}{16} & \frac{1}{16} & \frac{5}{16} & 0 & 0 & 0 & 0 & \frac{1}{16} & \frac{1}{16} & 0 & 0 & 0 & \frac{1}{16} & 0 & 0 & 0 & 0 & 0 \\
\frac{3}{40} & \frac{3}{40} & \frac{5}{16} & 0 & \frac{5}{16} & \frac{3}{40} & 0 & 0 & 0 & 0 & \frac{3}{40} & 0 & 0 & 0 & 0 & 0 & \frac{3}{40} & 0 & 0 & 0 \\
\frac{7}{160} & \frac{7}{160} & \frac{9}{16} & \frac{7}{160} & \frac{7}{160} & \frac{7}{160} & 0 & \frac{7}{160} & 0 & 0 & \frac{7}{160} & \frac{7}{160} & \frac{7}{160} & 0 & 0 & 0 & \frac{7}{160} & 0 & 0 & 0 \\
\frac{3}{40} & \frac{3}{40} & \frac{5}{16} & \frac{5}{16} & 0 & \frac{3}{40} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{3}{40} & 0 & 0 & 0 & 0 & \frac{3}{40} & 0 & 0 \\
\frac{7}{160} & \frac{7}{160} & \frac{7}{160} & \frac{9}{16} & \frac{7}{160} & \frac{7}{160} & 0 & 0 & \frac{7}{160} & 0 & 0 & 0 & \frac{7}{160} & \frac{7}{160} & \frac{7}{160} & 0 & 0 & \frac{7}{160} & 0 & 0 \\
\frac{3}{40} & \frac{3}{40} & 0 & \frac{5}{16} & \frac{5}{16} & \frac{3}{40} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{3}{40} & 0 & 0 & 0 & \frac{3}{40} & 0 \\
\frac{7}{160} & \frac{7}{160} & \frac{7}{160} & \frac{7}{160} & \frac{9}{16} & \frac{7}{160} & 0 & 0 & 0 & \frac{7}{160} & \frac{7}{160} & 0 & 0 & 0 & \frac{7}{160} & \frac{7}{160} & 0 & 0 & \frac{7}{160} & 0 \\
\frac{3}{40} & 0 & \frac{5}{16} & \frac{3}{40} & \frac{3}{40} & \frac{5}{16} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{3}{40} & \frac{3}{40} & 0 & 0 \\
\frac{3}{40} & 0 & \frac{3}{40} & \frac{5}{16} & \frac{3}{40} & \frac{5}{16} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{3}{40} & \frac{3}{40} & 0 \\
\frac{3}{40} & 0 & \frac{3}{40} & \frac{3}{40} & \frac{5}{16} & \frac{5}{16} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{3}{40} & 0 & \frac{3}{40} & 0 \\
\frac{7}{128} & 0 & \frac{7}{128} & \frac{7}{128} & \frac{7}{128} & \frac{9}{16} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{7}{128} & \frac{7}{128} & \frac{7}{128} & \frac{7}{128}
\end{array}\right)$$

The first 6 eigenvalues of the matrix $\mathbf{S}_v$ can be computed by the eigenvalues of the top-left 6×6 submatrix. The values are:

$$\{1, \frac{5}{16}, \frac{15 + \sqrt{70}}{80}, \frac{3}{20}, \frac{1}{8}, \frac{1}{8}\}.$$

B.2 Extraordinary Edge Subdivision Matrix

The matrix $\mathbf{S}_v$ below is the subdivision matrix for an extraordinary edge with the valence 4. The size of the matrix is 19×19. The major diagonlas are chosen in the similar way to the first case of Figure 23. The subdivision matrices for extraordinary vertices with different valences have the similar structure.

$$\mathbf{S}_e = \begin{pmatrix}
\frac{9}{16} & \frac{7}{160} & \frac{7}{160} & \frac{7}{160} & \frac{7}{160} & \frac{7}{160} & \frac{7}{160} & \frac{7}{160} & \frac{7}{160} & \frac{7}{160} & \frac{7}{160} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{5}{16} & \frac{3}{40} & 0 & \frac{5}{16} & \frac{3}{40} & 0 & \frac{3}{40} & \frac{3}{40} & \frac{3}{40} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{5}{16} & \frac{3}{40} & 0 & \frac{3}{40} & \frac{5}{16} & \frac{3}{40} & 0 & 0 & \frac{3}{40} & \frac{3}{40} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{5}{16} & \frac{3}{40} & 0 & 0 & \frac{3}{40} & \frac{5}{16} & \frac{3}{40} & 0 & 0 & \frac{3}{40} & \frac{3}{40} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{5}{16} & \frac{3}{40} & 0 & \frac{3}{40} & 0 & \frac{3}{40} & \frac{5}{16} & \frac{3}{40} & 0 & 0 & \frac{3}{40} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{5}{16} & 0 & \frac{3}{40} & \frac{3}{40} & 0 & 0 & \frac{3}{40} & \frac{5}{16} & \frac{3}{40} & 0 & \frac{3}{40} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{5}{16} & 0 & \frac{3}{40} & \frac{3}{40} & \frac{3}{40} & 0 & 0 & \frac{3}{40} & \frac{5}{16} & \frac{3}{40} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{5}{16} & 0 & \frac{3}{40} & 0 & \frac{3}{40} & \frac{3}{40} & 0 & 0 & \frac{3}{40} & \frac{5}{16} & \frac{3}{40} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{5}{16} & 0 & \frac{3}{40} & 0 & 0 & \frac{3}{40} & \frac{3}{40} & \frac{3}{40} & 0 & \frac{3}{40} & \frac{5}{16} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{16} & 0 & 0 & \frac{5}{16} & 0 & 0 & \frac{1}{16} & \frac{5}{16} & \frac{1}{16} & 0 & 0 & \frac{1}{16} & \frac{1}{16} & 0 & 0 & 0 & 0 & 0 & \frac{1}{16} \\
\frac{1}{8} & 0 & 0 & \frac{1}{4} & \frac{1}{8} & 0 & 0 & \frac{1}{8} & \frac{1}{4} & 0 & 0 & 0 & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{16} & 0 & 0 & \frac{1}{16} & \frac{5}{16} & 0 & 0 & 0 & \frac{5}{16} & \frac{1}{16} & 0 & 0 & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & 0 & 0 & 0 & 0 \\
\frac{1}{8} & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{8} & 0 & 0 & \frac{1}{8} & \frac{1}{4} & 0 & 0 & 0 & 0 & \frac{1}{8} & 0 & 0 & 0 & 0 \\
\frac{1}{16} & 0 & 0 & 0 & \frac{1}{16} & \frac{5}{16} & 0 & 0 & 0 & \frac{5}{16} & \frac{1}{16} & 0 & 0 & 0 & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & 0 & 0 \\
\frac{1}{8} & 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{8} & 0 & 0 & \frac{1}{8} & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & \frac{1}{8} & 0 & 0 \\
\frac{1}{16} & 0 & 0 & 0 & 0 & \frac{1}{16} & \frac{5}{16} & \frac{1}{16} & 0 & 0 & \frac{5}{16} & 0 & 0 & 0 & 0 & 0 & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} \\
\frac{1}{8} & 0 & 0 & \frac{1}{8} & 0 & 0 & \frac{1}{4} & \frac{1}{4} & 0 & 0 & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{8}
\end{pmatrix}$$

The first 11 eigenvalues computed by the top-left 11×11 submatrix is:

$$\{1, \frac{45+\sqrt{985}}{160}, \frac{25+6\sqrt{2}}{80}, \frac{25+6\sqrt{2}}{80}, \frac{19}{80}, \frac{25-6\sqrt{2}}{80}, \frac{25-6\sqrt{2}}{80}, \frac{3}{16}, \frac{13}{80}, \frac{13}{80}, \frac{1}{8}\}.$$

The different choices of the major diagonals gives a different subdivision matrix and the eigen-structure. If we choose all the diagonals toward the extraordinary edge, we acquire a different subdivision matrix:

$$\mathbf{S}'_e = \begin{pmatrix}
\frac{9}{16} & \frac{7}{160} & \frac{7}{160} & \frac{7}{160} & \frac{7}{160} & \frac{7}{160} & \frac{7}{160} & \frac{7}{160} & \frac{7}{160} & \frac{7}{160} & \frac{7}{160} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{5}{16} & \frac{1}{16} & 0 & \frac{5}{16} & \frac{1}{16} & 0 & \frac{1}{16} & \frac{1}{16} & 0 & 0 & 0 & 0 & \frac{1}{16} & 0 & 0 & 0 & 0 & 0 & \frac{1}{16} \\
\frac{5}{16} & \frac{1}{16} & 0 & \frac{1}{16} & \frac{5}{16} & \frac{1}{16} & 0 & 0 & \frac{1}{16} & 0 & 0 & 0 & \frac{1}{16} & 0 & \frac{1}{16} & 0 & 0 & 0 & 0 \\
\frac{5}{16} & \frac{1}{16} & 0 & 0 & \frac{1}{16} & \frac{5}{16} & \frac{1}{16} & 0 & 0 & \frac{1}{16} & 0 & 0 & 0 & 0 & \frac{1}{16} & 0 & \frac{1}{16} & 0 & 0 \\
\frac{5}{16} & \frac{1}{16} & 0 & \frac{1}{16} & 0 & \frac{1}{16} & \frac{5}{16} & 0 & 0 & 0 & \frac{1}{16} & 0 & 0 & 0 & 0 & 0 & \frac{1}{16} & 0 & \frac{1}{16} \\
\frac{5}{16} & 0 & \frac{1}{16} & \frac{1}{16} & 0 & 0 & 0 & \frac{5}{16} & \frac{1}{16} & 0 & \frac{1}{16} & 0 & \frac{1}{16} & 0 & 0 & 0 & 0 & 0 & \frac{1}{16} \\
\frac{5}{16} & 0 & \frac{1}{16} & 0 & \frac{1}{16} & 0 & 0 & \frac{1}{16} & \frac{5}{16} & \frac{1}{16} & 0 & 0 & \frac{1}{16} & 0 & \frac{1}{16} & 0 & 0 & 0 & 0 \\
\frac{5}{16} & 0 & \frac{1}{16} & 0 & 0 & \frac{1}{16} & 0 & 0 & \frac{1}{16} & \frac{5}{16} & \frac{1}{16} & 0 & 0 & 0 & \frac{1}{16} & 0 & \frac{1}{16} & 0 & 0 \\
\frac{5}{16} & 0 & \frac{1}{16} & 0 & 0 & 0 & \frac{1}{16} & \frac{1}{16} & 0 & \frac{1}{16} & \frac{5}{16} & 0 & 0 & 0 & 0 & 0 & \frac{1}{16} & 0 & \frac{1}{16} \\
\frac{1}{8} & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & \frac{1}{8} & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 & \frac{1}{8} \\
\frac{1}{4} & 0 & 0 & \frac{1}{8} & \frac{1}{8} & 0 & 0 & \frac{1}{8} & \frac{1}{8} & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{8} & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & 0 & 0 & 0 & 0 \\
\frac{1}{4} & 0 & 0 & 0 & \frac{1}{8} & \frac{1}{8} & 0 & 0 & \frac{1}{8} & \frac{1}{8} & 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 \\
\frac{1}{8} & 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & 0 & 0 \\
\frac{1}{4} & 0 & 0 & 0 & 0 & \frac{1}{8} & \frac{1}{8} & 0 & 0 & \frac{1}{8} & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 \\
\frac{1}{8} & 0 & 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \\
\frac{1}{4} & 0 & 0 & \frac{1}{8} & 0 & 0 & \frac{1}{8} & \frac{1}{8} & 0 & 0 & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{4}
\end{pmatrix}$$

In this case, there is no simple reordering of the vertices in $\mathbf{p}^0$ that gives the block structure. However, we can still compute all the first 11 eigenvalues from $\mathbf{S}'_e$:

$$\{1, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{25+\sqrt{385}}{160}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{8}\}.$$

As previously mentioned, the difference between the eigenvalues of the subdivision matrices due to the choice of the major diagonals are subtle if the edge valence is low. However, if the edge valence becomes higher, in particular over 8, the difference in the eigenstructure of the subdivision matrices become larger.

References

Bajaj, C., Warren, J., Xu, G., 2002. A subdivision scheme for hexahedral meshes. The Visual Computer 18, 343–356.

Boehm, W., 1984. Calculating with box splines. Computer Aided Geometric Design 1 (2), 149–162.

Catmull, E., Clark, J., Sep. 1978. Recursively generated B-spline surfaces on arbitrary topological meshes. Computer-Aided Design 10, 350–355.

Chaikin, G., 1974. An algorithm for high speed curve generation. Computer Graphics and Image Processing 3, 346–349.

Chang, Y.-S., McDonnell, K. T., Qin, H., 2002. A new solid subdivision scheme based on box splines. In: Proceedings of Solid Modeling 2002. pp. 226–233.

Chang, Y.-S., McDonnell, K. T., Qin, H., May 2003. An interpolatory subdivision for volumetric models over simplicial complexes. In: Proceedings of Shape Modeling International 2003. pp. 143–152.

Chang, Y.-S., Qin, H., 2004. A framework for multi-dimensional adaptive subdivision objects. In: Proceedings of Solid Modeling 2004. pp. 123–134.

Coxeter, H. S. M., 1963. Regular Polytopes, 2nd Edition. The Macmillan Company, New York.

de Boor, C., Höllig, K., Riemenschneider, S., 1993. Box Splines. Springer-Verlag, New York.

Doo, D., Sabin, M., Sep. 1978. Behaviour of recursive division surfaces near extraordinary points. Computer-Aided Design 10 (6), 356–360.

Dyn, N., Gregory, J., Levin, D., 1991. Analysis of uniform binary subdivision schemes for curve design. Constructive Approximation 7 (2), 127–148.

Dyn, N., Levin, D., Gregory, J., Apr. 1990. A butterfly subdivision scheme for surface interpolation with tension control. ACM Transactions on Graphics 9 (2), 160–169.

Dyn, N., Levin, D., Liu, D., Apr. 1992. Interpolatory convexity-preserving subdivision schemes for curves and surfaces. Computer-Aided Design 24(4), 211–216.

Dyn, N., Micchelli, C. A., 1990. Using parameters to increase smoothness of curves and surfaces generated by subdivision. Computer Aided Geometric Design 7, 129–140.

General Electric Company, 2004. 3.0T Signa EXCITE HD. *http://www.gehealthcare.com/usen/mr*.

Ghosh, A., Miyamoto, Y., Reimanis, I., 1997. Functionally graded materials: Manufacture, properties and applications. Ceramic Transactions 76.

Hoppe, H., DeRose, T., Duchamp, T., Halstead, M., Jin, H., McDonald, J., Schweitzer, J., Stuetzle, W., 1994. Piecewise smooth surface reconstruction. In: Computer Graphics (SIGGRAPH ’94 Conference Proceedings). pp. 295–302.

Kobbelt, L., 1996a. Interpolatory subdivision on open quadrilateral nets with arbitrary topology. In: Computer Graphics Forum (Proceedings of Eurographics ’96). pp. 409–420.

Kobbelt, L., Nov. 1996b. A variational approach to subdivision. Computer Aided Geometric Design 13 (8), 743–761.

Levin, A., 1999. Combined subdivision schemes for the design of surfaces satisfying boundary conditions. Computer Aided Geometric Design 16 (5), 345–354.

Levin, A., Levin, D., 2003. Analysis of quasi uniform subdivision. Applied and Computational Harmonic Analysis 15 (1), 18–32.

Linsen, L., Pascucci, V., Duchaineau, M. A., Hamann, B., Joy, K. I., 2002. Hierarchical representation of time-varying volume data with $sqrt[4]2$ subdivision and quadrilinear b-spline wavelets. In: Proceedings of Pacific Graphics 2002. pp. 346–355.

Loop, C., 1987. Smooth subdivision surfaces based on triangles. Master's thesis, University of Utah, Department of Mathematics.

MacCracken, R., Joy, K. I., 1996. Free-Form deformations with lattices of arbitrary topology. In: Computer Graphics (SIGGRAPH '96 Conference Proceedings). pp. 181–188.

McDonnell, K. T., Qin, H., May 2000. Dynamic sculpting and animation of free-form subdivision solids. In: Proceedings of IEEE Computer Animation 2000. pp. 126–133.

Micchelli, C., Prautzsch, H., 1987. Computing surfaces invariant under subdivision. Computer Aided Geometric Design 4 (4), 321–328.

Miyamoto, Y., Kaysser, W. A., Rabin, B. H., 1999. Functionally Graded Materials: Design, Processing and Applications. Kluwer Academic, Boston.

Pascucci, V., Sep. 2002. Slow growing subdivision (sgs) in any dimension: Towards removing the curse of dimensionality. Computer Graphics Forum (Proceeding of Eurographics 2002) 21 (3), 451–460.

Prautzsch, H., 1985. Generalized subdivision and convergence. Computer Aided Geometric Design 2 (1-3), 69–76.

Prautzsch, H., 1998. Smoothness of subdivision surfaces at extraordinary points. Advances in Computational Mathematics 9, 377–390.

Prautzsch, H., Reif, U., 1997. Necessary conditions for subdivision surfaces. Report 97/04, Sonderforschungsbereich 404, Universitat Stuttgart.

Prautzsch, H., Reif, U., 1999. Degree estimates for C^k-piecewise polynomial subdivision surfaces. Advances in Computational Mathematics 10 (2), 209–217.

Prautzsch, H., Umlauf, G., 1998. A G^2-subdivision algorithm. In: Proceedings of the Dagstuhl conference on geometric modeling 1996, Computing suppl. pp. 217–224.

Reif, U., 1995a. Some new results on subdivision algorithms for meshes of arbitrary topology. Approximation Theory VIII 2, 367–374.

Reif, U., 1995b. A unified approach to subdivision algorithms near extraordinary vertices. Computer Aided Geometric Design 12, 153–174.

Requicha, A. A. G., Voelcker, H. B., Mar. 1982. Solid modeling: a historical summary and contemporary assessment. IEEE Computer Graphics and Applications 2, 9–23.

Santos, F., Seidel, R., Apr. 2003. A better upper bound on the number of triangulations of a planar point set. Journal of Combinatorial Theory, Series A 102 (1), 186–193.

Schaefer, S., Hakenberg, J., Warren, J., 2004. Smooth subdivision of tetrahedral meshes. In: Eurographics Symposium on Graphics Processing 2004. pp. 151–158.

Warren, J., Weimer, H., 2001. Subdivision Methods for Geometric Design: A constructive Approach. Morgan Kaufmann Publisher.

Zagajac, J., Mar. 1996. A fast method for estimating discrete field values in early engineering design. IEEE Transactions on Visualization and Computer Graphics 2 (1), 35–43.

Zorin, D., 1997. Subdivision and multiresolution surface representations. Ph.D. thesis, Caltech, Pasadena.

Zorin, D., 2000. Smoothness of stationary subdivision on irregular meshes. Constructive Approximation 16, 359–398.

Zorin, D., Kristjansson, D., 2002. Evaluation of piecewise smooth subdivision surfaces. The Visual Computer 18, 299–315.

Zorin, D., Schröder, P., Sweldens, W., 1996. Interpolating subdivision for meshes with arbitrary topology. In: Computer Graphics (SIGGRAPH ’96 Conference Proceedings). pp. 189–192.