On the linear independence of (truncated) hierarchical subdivision splines

Zore, U; Jüttler, B; Kosinka, Jiri

Published in:
Geometry+Simulation Report

Publication date:
2015

NFN - Nationales Forschungsnetzwerk

Geometry + Simulation

http://www.gs.jku.at

On the Linear Independence of (Truncated) Hierarchical Subdivision Splines

U. Zore, B. Jüttler, J. Kosinka

G+S Report No. 40
This technical report replaces report no. 17.

November 2015

Der Wissenschaftsfonds.

JOHANNES KEPLER UNIVERSITY LINZ

On the Linear Independence of (Truncated) Hierarchical Subdivision Splines

Urška Zore[a], Bert Jüttler[a,*], Jiří Kosinka[b]

[a]*Institute of Applied Geometry, Johannes Kepler University, Linz, Austria*
[b]*Johann Bernoulli Institute, University of Groningen, The Netherlands*

Abstract

Motivated by the necessity to perform adaptive refinement in geometric design and numerical simulation, (truncated) hierarchical generating systems for nested spaces of spline functions defined on domains in $\mathbb{R}^d$ have been recently introduced. Their linear independence can be guaranteed with the help of the local linear independence of the spline basis at each level. The present paper extends this framework to spline functions that are defined on domain manifolds, in particular focusing on the case of subdivision splines generated by the Catmull-Clark, Loop, and modified Butterfly subdivision schemes. Since the property of local linear independence is no longer available, we introduce the concept of safe subdomains, which allows us to guarantee linear independence. We provide a catalog of safe subdomains that facilitates the design of domain hierarchies with linearly independent (truncated) hierarchical generating systems.

Keywords: Multi-level spline space, generating system, subdivision scheme, Catmull-Clark subdivision, Loop subdivision, Butterfly subdivision

1. Introduction

The powerful framework of Isogeometric Analysis [4] facilitates the exchange of data between various software tools used for geometric design (CAD systems) and for analysis (numerical simulations). The use of B-splines and NURBS not only for modeling but also for analysis offers advantages over traditional finite element functions, such as increased smoothness, faster convergence, improved stability, and most importantly, it eliminates the need for model (re)meshing.

However, since multivariate spline representations are based on tensor-product constructions, they suffer from two major limitations: local refinement is not possible and only trivial box-like topologies are supported. Various generalizations of tensor-product splines, such as T-splines, hierarchical splines, LR splines, and PHT splines, have been introduced in order to facilitate *local adaptivity*.

T-splines [17, 27] are splines defined by local knot vectors. Hierarchical B-splines [33] are obtained by combining selected B-splines from a sequence of nested spline spaces. LR splines [12] are constructed by repeatedly splitting tensor-product B-splines, starting from

*Corresponding author. Tel. +43(0)732 2468 4080
Email addresses: urska.zore@jku.at (Urška Zore), bert.juettler@jku.at (Bert Jüttler), j.kosinka@rug.nl (Jiří Kosinka)

an initial set defined on a mesh of tensor-product topology. PHT splines [16] are based on the full space of piece-wise polynomial functions on a given T-mesh, which is equipped with a suitable basis.

In the context of this paper, we are particularly interested in (truncated) hierarchical B-splines (THB-splines) [9, 10]. This spline basis provides various useful properties in the mathematical framework of hierarchical splines, such as partition of unity, strong stability, full approximation power, and efficient implementation [13, 31]. This approach has recently been extended to more general hierarchies of spline spaces [36].

As the domains of tensor-product spline functions are boxes in parameter space, it becomes more challenging to create suitable representations for domains of *general manifold topology*. A typical solution consists in using multi-patch representations with reduced smoothness across their interfaces [14, 28]. This may, however, lead to artifacts in numerical solutions at the interfaces. Moreover, global high-order smoothness is advantageous for solving higher-order problems, such as the biharmonic equation.

The framework of T-splines [29] includes extraordinary vertices (i.e., vertices where other than four patches meet), which are essential for dealing with general topologies, but the mathematical properties of the isogeometric functions in the vicinity of these points are not well understood.

Typically, the presence of extraordinary vertices leads to a reduced order of smoothness, approximation, and convergence rate [20]. Nevertheless, the use of subdivision functions seems to be one of the most promising approaches [1, 3, 11], mainly due to their support for arbitrary manifold topology, in-built refinement relations, and the widespread use of subdivision representations in applications, especially in Computer Graphics [6].

Using trimmed NURBS representations is a different approach, which is adopted by CAD systems. When used in isogeometric analysis, these representations require special treatment [26]. Alternatively it is possible to convert them to subdivision surfaces [30].

The present paper describes an extension of the THB construction to functions generated by subdivision algorithms. It combines adaptive refinement with the topological flexibility of subdivision. The concept of a domain manifold [23] is employed to define subdivision splines in a mathematically rigorous way. Based on this framework, we derive conditions for linear independence of (truncated) hierarchical subdivision blending functions.

Subdivision algorithms create sequences of nested spaces and are therefore suited for invoking the multi-resolution framework, e.g., when performing hierarchical editing [19]. This is in fact analogous to hierarchical B-spline refinement, which was originally formulated by Forsey and Bartels [8]. Their construction was later augmented by introducing a *basis* [15], thereby facilitating its application in adaptive surface reconstruction and isogeometric analysis. It is desirable to introduce a similar basis for hierarchical subdivision splines.

While preparing this manuscript, we became aware of the recent papers [34, 35], which report on a parallel development of another group of authors. These papers focus on numerical simulation with truncated hierarchical Catmull-Clark subdivision functions and provide sufficient conditions for linear independence of the corresponding basis functions.

In contrast, the present paper formulates the hierarchical construction in a more general framework, which encompasses, among other schemes, Catmull-Clark and Loop subdivision surfaces, and surfaces constructed by the interpolatory (modified) Butterfly subdivision

scheme. We discuss the linear independence of the hierarchical generating system in a mathematically rigorous way. More precisely, we introduce catalogs of *safe subdomains* to describe sufficient conditions on the domain hierarchy that ensure linear independence.

In order to obtain these catalogs we use techniques that were pioneered by Peters and Wu [24] when discussing linear independence for Catmull-Clark and Loop subdivision. We cast these techniques into a general framework that allows to derive catalogs of safe subdomains for a wide range of subdivision schemes using a unified approach.

The remainder of the paper is organized as follows. Section 2 extends the construction of (truncated) hierarchical generating systems presented in [36] to spaces of functions defined on manifolds and to generating systems containing functions that are not necessarily non-negative. The next section introduces the concept of safe subdomains to establish sufficient conditions for linear independence. Section 4 introduces (truncated) hierarchical subdivision splines. Section 5 shows how to identify safe subdomains that are needed to guarantee linear independence, summarizes the results known for Catmull-Clark and Loop subdivision schemes, and additionally analyzes the case of Butterfly subdivision blending functions. Finally, we conclude the paper.

2. Hierarchical generating systems on manifolds

The framework of hierarchical generating systems was developed in [36] for non-negative functions in $C(\Omega)$, where the domain Ω was required to be an open subset of $\mathbb{R}^d$. Local linear independence was used to certify their linear independence.

In order to generalize this concept to subdivision splines, we extend this framework in two ways: Firstly, we consider functions defined on d-dimensional manifolds, and secondly, we allow functions that are not necessarily non-negative. Furthermore, we identify sufficient conditions that ensure linear independence of hierarchical generating systems. These conditions are weaker than local linear independence.

2.1. Functions on manifolds

Spaces of functions defined on an open set $\Omega \subset \mathbb{R}^d$ that are spanned by finite *generating systems* G were considered in [36]. In the following, we generalize the domain to a d-dimensional manifold $\mathcal{M}$.

A d-dimensional manifold $\mathcal{M}$ is a topological space that locally resembles the d-dimensional Euclidean space, i.e., each point has a neighborhood homeomorphic to the Euclidean space of dimension d. More precisely, each *inner* point of the manifold has a neighborhood homeomorphic to an open ball in $\mathbb{R}^d$,

$$\{(x_1, \dots, x_d) \in \mathbb{R}^d \mid \textstyle\sum_i x_i^2 < 1\},$$

and each point on its *boundary*, which can be empty, has a neighborhood homeomorphic to a semi-open half-ball in $\mathbb{R}^d$,

$$\{(x_1, \dots, x_d) \in \mathbb{R}^d \mid \textstyle\sum_i x_i^2 < 1 \text{ and } x_1 \geq 0\},$$

where all boundary points correspond to points with $x_1 = 0$. A d-dimensional manifold with or without boundary will be referred to as *d-manifold*.

The open unit square $(0, 1)^2$ is a 2-manifold without boundary, while the closed unit square $[0, 1]^2$ is a 2-manifold with boundary. Curves and surfaces in $\mathbb{R}^3$ can be equipped

with a manifold structure, provided that they do not possess self-intersections or singularities. However, the notion of manifolds also encompasses more abstract objects, some of which will be described later in Section 4.

Given a d-manifold $\mathcal{M}$, any mapping $\mathcal{M} \to \mathbb{R}$ is called a function on $\mathcal{M}$. In this context, we say that $\mathcal{M}$ is the *domain manifold* of this function. Due to the topological structure of $\mathcal{M}$ it is possible to extend the notion of *continuity* to functions on manifolds, and we consider only continuous functions in the remainder of this paper.

We define the *support* of a function f on $\mathcal{M}$, denoted by $\operatorname{supp} f$, as the set of all points of $\mathcal{M}$ where the function is non-zero. The support of a continuous function is an open d-submanifold of $\mathcal{M}$.

2.2. Hierarchical generating systems

A *generating system* G on a domain manifold $\mathcal{M}$ is a finite system of continuous functions defined on $\mathcal{M}$. We express G as the column vector

$$G = (\gamma_i)_{i \in \mathcal{I}},$$

indexed by a finite set $\mathcal{I}$. We will use row vectors $c = (c_i)_{i \in \mathcal{I}}^T$ to collect the coefficients of the functions in

$$\mathcal{G} = \operatorname{span} G = \mathbb{R}^{\mathcal{I}} G = \{c\, G \mid c \in \mathbb{R}^{\mathcal{I}}\}.$$

The Cartesian product of m copies of $\mathcal{G}$ gives rise to the space $\mathcal{G}^m$, which consists of geometric realizations of the manifold $\mathcal{M}$ in the m-dimensional real space. It should be noted that these realizations are not necessarily manifolds themselves, since they may have self-intersections.

In contrast to [36], we do not restrict ourselves to non-negative functions. Generating systems that partition unity, i.e.,

$$\mathbf{1} G = 1$$

with $\mathbf{1} = (1, \ldots, 1) \in \mathbb{R}^{\mathcal{I}}$, are said to be *normalized*.

We consider an infinite sequence of generating systems with a refinement property. More precisely, we consider generating systems $G^\ell = (\gamma_i^\ell)_{i \in \mathcal{I}^\ell}$, $\ell = 0, 1, \ldots$, with index sets $\mathcal{I}^\ell$, where the additional index ℓ is called the *level*. We assume that there exist matrices R^ℓ such that the generating systems satisfy the refinement equation

$$G^\ell = R^{\ell+1} G^{\ell+1}, \tag{1}$$

where the columns of the matrices $R^{\ell+1}$ sum to one. Each function of G^ℓ is expressed as a linear combination of functions of level $\ell + 1$. Again, in contrast to [36], the entries of the matrices are not assumed to be non-negative.

Hierarchical generating systems can now be defined with the help of a *subdomain hierarchy* that contains $N + 1$ levels, where N is a non-negative integer. The subdomain hierarchy is a decreasing sequence $(\mathcal{M}^\ell)_{\ell=0,\ldots,N+1}$ of $N + 2$ d-dimensional submanifolds satisfying

$$\mathcal{M} = \mathcal{M}^0 \supseteq \mathcal{M}^1 \supseteq \cdots \supseteq \mathcal{M}^N \supseteq \mathcal{M}^{N+1} = \emptyset, \tag{2}$$

where the submanifold $\mathcal{M}^{N+1} = \emptyset$ is introduced to simplify the notation later on. More precisely, each $\mathcal{M}^\ell$ is itself assumed to be a d-manifold with respect to the topology inherited from $\mathcal{M}$. Here we use $\supseteq$ to express the submanifold relation between manifolds.

The *differences* between the domain manifold $\mathcal{M}$ and the subdomains $\mathcal{M}^\ell$ define the complementary hierarchy of *rings*

$$\Delta^\ell = \mathcal{M} \setminus \mathcal{M}^{\ell+1}$$

satisfying

$$\Delta^0 \subseteq \Delta^1 \subseteq \cdots \subseteq \Delta^N = \mathcal{M}.$$

Conceptually, the ring Δ^ℓ is the entire domain manifold with a "hole" defined by $\mathcal{M}^{\ell+1}$. Note that the rings are not necessarily submanifolds.

According to the domain hierarchy we now select the functions γ_i^ℓ from each generating system G^ℓ that satisfy

$$\operatorname{supp} \gamma_i^\ell \subseteq \mathcal{M}^\ell \ \wedge \ \operatorname{supp} \gamma_i^\ell \nsubseteq \mathcal{M}^{\ell+1}.$$

The *selected* functions are collected in vectors $\hat{G}^\ell = (\gamma_i^\ell)_{i \in \hat{\mathcal{I}}^\ell}$, where the index set $\hat{\mathcal{I}}^\ell$ contains their indices. More precisely, $\hat{G}^\ell$ collects all the functions from G^ℓ whose support is contained in the subdomain $\mathcal{M}^\ell$ but is not contained in $\mathcal{M}^{\ell+1}$.

The *hierarchical generating system* is now obtained by collecting all subsets $\hat{G}^\ell$,

$$\mathcal{K} = \left(\hat{G}^\ell\right)_{\ell=0,\ldots,N} = \left(\gamma_i^\ell\right)_{i \in \hat{\mathcal{I}}^\ell, \ell=0,\ldots,N}. \tag{3}$$

The symbol $\mathcal{K}$ was chosen in reference to the inventor of this selection mechanism [15].

2.3. Nested hierarchical refinement and truncation

In order to be useful in practice, the adaptive refinement using hierarchical generating systems should produce nested spaces. More precisely, if one starts from a given subdomain sequence, then enlarging all subdomains should produce a hierarchical generating system which spans a space that contains the previous one. This also includes the case of adding a new level of refinement, since one may see this as enlarging a previously empty subdomain at the finest level while keeping the remaining subdomains unchanged.

In addition, it is desirable to obtain a generating system that forms a partition of unity, since this is an essential property for geometric modeling. The hierarchical generating system $\mathcal{K}$ does not possess this property. We introduce an assumption on the domain hierarchy that ensures nested refinement and allows us to invoke the truncation mechanism for restoring normalization.

We say that a function γ_i^ℓ *refines to* $\gamma_j^{\ell+1}$ if the corresponding entry $R_{ij}^{\ell+1}$ in the refinement matrix is non-zero. Note that the refinement matrix is not necessarily unique since we do not assume linear independence of the generating systems. In the case of non-unique refinement matrices, the relation refers to a certain fixed choice. We assume that the domain hierarchy satisfies:

$$\forall i \in \mathcal{I}^\ell \, \forall j \in \mathcal{I}^{\ell+1} \colon \ \operatorname{supp} \gamma_i^\ell \subseteq \mathcal{M}^{\ell+1} \wedge R_{ij}^{\ell+1} \neq 0 \ \Rightarrow \ \operatorname{supp} \gamma_j^{\ell+1} \subseteq \mathcal{M}^{\ell+1}. \tag{4}$$

In other words, if the subdomain $M^{\ell+1}$ contains the support of a function of level ℓ, then it also contains the supports of all the functions of the next level it refines to. Consequently, if a function of level ℓ is not selected for the hierarchical generating system, even though its support is contained in $\mathcal{M}^\ell$, then it can be represented as a linear combination of functions that are selected at higher levels. It should be noted that this assumption is

always satisfied for non-negative generating systems and refinement matrices with only non-negative entries, as considered in [36].

Under the assumption of (4), the hierarchical refinement creates *nested hierarchical spaces*. More precisely, enlarging the subdomains always creates a hierarchical generating system that spans a superspace of the previous one, as described in [33, Proposition 4].

Moreover, this assumption enables us to restore partition of unity by the truncation mechanism. More precisely, the truncation with respect to level $\ell+1$ is defined by

$$\operatorname{trunc}^{\ell+1}(c^\ell G^\ell) = c^\ell \hat{R}^{\ell+1} G^{\ell+1}, \tag{5}$$

where the *reduced refinement matrix* $\hat{R}^{\ell+1}$ is obtained by replacing the columns that correspond to selected functions from level $\ell+1$ by zeros. We define the *truncated hierarchical generating system*

$$\mathcal{T} = (\operatorname{trunc}^N(\cdots \operatorname{trunc}^{\ell+1}(\gamma_i^\ell)\cdots))_{i\in\hat{\mathcal{I}}^\ell,\ell=0,\dots,N}, \tag{6}$$

which forms a partition of unity if all generating systems G^ℓ are normalized. This can be proved by generalizing the results in [36]. Note that the truncation does not affect the values of functions of level ℓ restricted to the ring Δ^ℓ, i.e.,

$$\operatorname{trunc}^N(\cdots \operatorname{trunc}^{\ell+1}(\gamma_i^\ell)\cdots)|_{\Delta^\ell} = \gamma_i^\ell|_{\Delta^\ell} \text{ for } i \in \hat{\mathcal{I}}^\ell,\ \ell = 0,\dots,N.$$

3. Linear independence of hierarchical generating systems

It has been shown that *local linear independence* of the generating systems G^ℓ on $\Omega \subset \mathbb{R}^d$ is a sufficient condition for the linear independence of the hierarchical generating system [9, 36]. This observation can be extended to functions defined on domain manifolds. Local linear independence, however, is a relatively strong condition, which is not satisfied in certain important applications, such as functions defined by subdivision schemes. The analysis of weaker sufficient conditions is therefore of interest. This can be done by formulating conditions on the subdomain hierarchy.

3.1. Safe subdomains

We introduce the notion of *G^ℓ-safe subdomains* and show that these can be used to guarantee linear independence, provided that the subdomain hierarchy satisfies certain conditions.

Definition 1. A subdomain S is said to be *G^ℓ-safe* if the functions

$$\{\gamma|_S \mid \gamma \in G^\ell \ \text{ and } \ \gamma|_S \neq 0\}$$

are linearly independent.

Obviously, the generating system G^ℓ is linearly independent if the entire domain Ω is G^ℓ-safe. Furthermore, *local* linear independence of the generating system G^ℓ is characterized by the fact that *every* open subset $S \subset \Omega$ is G^ℓ-safe. Since subdivision splines do not always possess the latter property, we consider linear independence only on certain subdomains, and the notion of *safe* subdomains conveniently describes this property.

The following two observations are implied by this definition.

Lemma 2. *The union of G^ℓ-safe subdomains is G^ℓ-safe. Moreover, if S is a subdomain and there exists a G^ℓ-safe subdomain $S' \subset S$ such that the functions in G^ℓ that are non-zero on S are exactly the ones that are non-zero on S', i.e.,*

$$\forall \gamma \in G^\ell : \ \gamma|_S \neq 0 \ \Rightarrow \ \gamma|_{S'} \neq 0,$$

then S is also G^ℓ-safe.

In particular, the second part of the lemma implies that the closure of a G^ℓ-safe subdomain is again G^ℓ-safe, since we consider continuous functions only. Note that the intersection of G^ℓ-safe subdomains is not necessarily G^ℓ-safe.

Note that local linear independence of G^ℓ on $\mathcal{M}$ implies that every open subdomain is G^ℓ-safe. If the generating systems of all levels ℓ possess this property, then the linear independence of $\mathcal{K}$ is automatically guaranteed, without additional assumptions about the subdomain hierarchy. In fact, all rings Δ^ℓ are then closures of safe subdomains and therefore G^ℓ-safe, cf. Lemma 2.

All subdomains that we identify as G^ℓ-safe are collected in a *catalog* of certified G^ℓ-safe subdomains, denoted by C^ℓ and formally understood as a set. Given a G^ℓ-safe subdomain U, we denote with

$$[U]_\ell = \mathcal{M} \setminus \bigcup_{\gamma \in G^\ell, \gamma|_U = 0} \operatorname{supp} \gamma \tag{7}$$

the maximal subdomain $[U]_\ell \supseteq U$ that is known to be G^ℓ-safe according to the second part of Lemma 2. Note that any set U' satisfying $U \subseteq U' \subseteq [U]_\ell$ is also G^ℓ-safe.

3.2. Ensuring linear independence

We formulate a sufficient condition for linear independence:

Theorem 3. *If each ring Δ^ℓ is G^ℓ-safe, $\ell \in \{0, \dots, N\}$, then the (truncated) hierarchical generating system $\mathcal{K}$ (or $\mathcal{T}$) is linearly independent.*

Proof. We need to show that

$$b\,\mathcal{K} = 0 \ \text{ implies } \ b = \mathbf{0}, \tag{8}$$

where b is a row vector of coefficients and $\mathbf{0}$ is a row null vector of the same dimension. Following [15] and [9], we decompose and rearrange the left-hand side of (8) according to the hierarchy of the generating systems,

$$b^0 \hat{G}^0 \ + \ \dots \ + \ b^N \hat{G}^N = 0. \tag{9}$$

The vector b^ℓ collects the coefficients of functions in $\hat{G}^\ell$.

The functions in the first term in (9) are the only non-zero functions acting on the ring Δ^0. By assumption, Δ^0 is G^0-safe and thus $\hat{G}^0$ is linearly independent on Δ^0. It follows that $b^0 = 0$.

In the remaining sum

$$b^1 \hat{G}^1 \ + \ \dots \ + \ b^N \hat{G}^N = 0,$$

only the functions in the first term are non-zero on the ring Δ^1. Consequently, the above argument can be used repeatedly, eventually exhausting all the terms in (9). Moreover, since truncation does not change the values of the functions on the corresponding rings, this proof applies to the truncated hierarchical generating system as well. □

Example 4. Zwart-Powell (ZP) elements were first introduced in [38], see also [5]. The hierarchical construction of the hierarchical Zwart-Powell generating system was analyzed in [36], in particular focusing on its linear dependencies. For each level, the generating system consists of the translates of a box-spline on a type-II triangulation. The box splines within each level are linearly dependent. A possibility to restore linear independence of the hierarchical generating system consists in removing one of the ZP elements from each level. Consider a linearly independent generating system G^ℓ of ZP elements, where one box spline has been discarded. Any open subdomain that possesses a non-empty intersection with the support of the discarded ZP element is G^ℓ-safe. Consequently, if the subdomain hierarchy $(\mathcal{M}^\ell)_{\ell=0,\dots,N}$ has been chosen such that any connected component of the rings Δ^ℓ has a non-empty intersection with the support of the discarded function of level ℓ, then linear independence of the hierarchical generating system is guaranteed.

Theorem 3 leads to a simple *refinement algorithm* that maintains linear independence of (truncated) hierarchical generating systems created by adaptive refinement, provided that we have a catalog of safe subdomains for the generating system G^ℓ on every level at hand. More precisely, assume we have a domain hierarchy (2) that satisfies the assumptions of Theorem 3 and let Z be a region marked for further refinement. We have to define a new domain hierarchy that still satisfies (2) and the assumptions of Theorem 3 so that the level of each point in Z (i.e., the largest ℓ so that $\mathcal{M}^\ell$ contains it) is increased. The new hierarchy contains $N+3$ subdomains satisfying

$$\mathcal{M} = \tilde{\mathcal{M}}^0 \supseteq \tilde{\mathcal{M}}^1 \supseteq \cdots \supseteq \tilde{\mathcal{M}}^{N+1} \supseteq \tilde{\mathcal{M}}^{N+2} = \emptyset,$$

which are defined recursively, starting at the finest level, by

$$\tilde{\mathcal{M}}^\ell = \big\langle \mathcal{M}^\ell \cup \tilde{\mathcal{M}}^{\ell+1} \cup (\mathcal{M}^{\ell-1} \cap Z) \big\rangle_{\ell-1}, \quad \ell = N+1, \dots, 1.$$

The operation $\langle\cdot\rangle_\ell$, which restores the G^ℓ-safety of the rings, is defined by $\langle V\rangle_\ell = \mathcal{M} \setminus W_V$, where

$$W_V = \Big[\bigcup_{\mathcal{M}\setminus V \supseteq T \in C^\ell} T \Big]_\ell \cap (\mathcal{M} \setminus V)$$

is the maximal subdomain contained in $\mathcal{M} \setminus V$ that is known to be G^ℓ-safe, cf. Lemma 2 and (7). An example of applying the algorithm in the case of Loop and Butterfly subdivision will be presented at the end of Section 5.

4. Hierarchical subdivision splines

We apply the theory of the previous section to spaces spanned by subdivision splines. In the family of approximating subdivision algorithms, we focus on the Catmull-Clark and Loop schemes. Among interpolatory subdivision algorithms, we consider the modified Butterfly scheme (further referred to as simply the Butterfly scheme), introduced in [37] as an improved version of the original scheme presented in [7]. Note that a more recent, non-stationary variant of the modified Butterfly scheme appeared in [22]. While Butterfly subdivision is defined for triangular meshes, interpolatory algorithms for quadrilateral meshes have been studied as well, see [21] and the references therein.

We assume that a (Catmull-Clark, Loop or Butterfly) subdivision surface is given by its control mesh. This control mesh is used to define an extended domain manifold, which

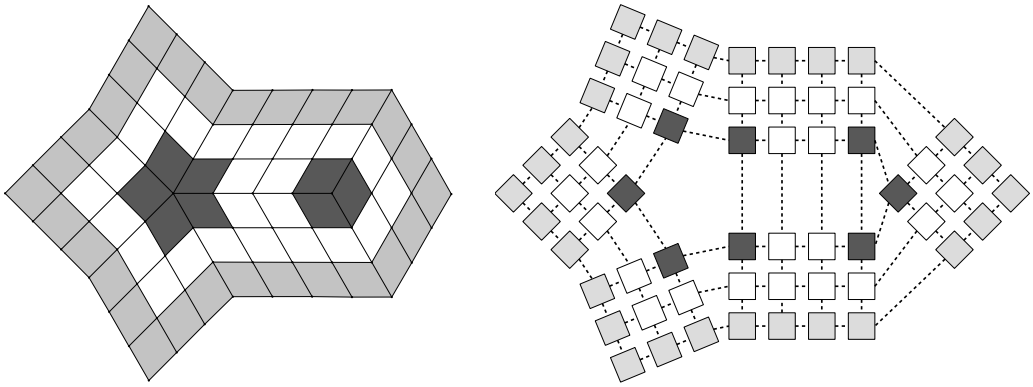


Figure 1: The extended domain manifold $\mathcal{E}^0$ (right) given by the control mesh of a Catmull-Clark subdivision surface (left). We distinguish between boundary (light gray), regular (white), and irregular cells (dark gray). The domain manifold $\mathcal{M}$ contains only the fundamental cells that correspond to inner cells (white and dark gray).

in turn gives the domain manifold $\mathcal{M}$ for the hierarchical generating system by omitting certain cells along the boundary, see below. The extended domain manifold is introduced in order to deal with subdivision surfaces with boundaries.

4.1. Domain manifolds

The starting point of our construction is the *mother cell* $\Diamond$, which is either the unit square $\square = [0,1]^2$ or the standard triangle

$$\triangle = \{(x,y) \mid x,y \geq 0 \ \text{ and } \ x+y \leq 1\} \subset \mathbb{R}^2.$$

We choose a finite index set $\mathcal{J} \subset \mathbb{Z}$ and consider the set $\Diamond \times \mathcal{J}$ as the union of the *fundamental cells* $\Diamond \times \{i\}$, $i \in \mathcal{J}$. Each fundamental cell is a copy of the mother cell, equipped with an index. Similarly to [23], which focuses on quadrilateral meshes (i.e., $\Diamond = \square$), we define a manifold with the help of a *neighbor relation* $\simeq$ on the set $\Diamond \times \mathcal{J}$. This relation identifies points on the edges of different fundamental cells,

$$((x,y),i) \ \simeq \ ((\tilde{x},\tilde{y}),j) = (\alpha_{ij}(x,y),j), \quad (x,y),(\tilde{x},\tilde{y}) \in \partial\Diamond, \tag{10}$$

where α_{ij} is an affine map from one edge of $\Diamond$ to another. For instance, if $\Diamond = \square$, the edge $(x,0)$ of the fundamental cell $\square \times \{i\}$ could be identified with the edge $(1,\tilde{y})$ of the cell $\square \times \{j\}$ via $\alpha(x,y) = (1,1-x)$.

If the cells with indices i and j are glued together, then we use the same map α_{ij} for all points on the common edge. Not all pairs of cells, however, are glued together. The relation $\simeq$ is derived from the connectivity of the control mesh of the subdivision surface.

Example 5. The control mesh of a Catmull-Clark subdivision surface with boundary (left) and the corresponding extended domain manifold (right) is shown in Figure 1. The 60 quadrilateral faces of the control mesh correspond to the 60 fundamental cells of the extended domain manifold. Two edges of fundamental cells are glued together along boundary edges (indicated by dashed lines) if the corresponding faces of the control mesh share an edge. Thus, each inner edge of the control mesh corresponds to one of the neighbor relations (10).

The *extended neighbor relation* $\hat{\simeq}$ is the transitive and reflexive closure of $\simeq$. We say that two fundamental cells are *neighbors* if at least one pair of points on their boundaries is related by $\hat{\simeq}$. Note that this includes vertex-vertex contacts of fundamental cells, since we consider the transitive closure of the neighbor relation $\simeq$.

When restricted to the corners of the fundamental cells, the extended neighbor relation induces an equivalence relation between them. The set of equivalence classes will be denoted by V^0. Each one of these equivalence classes will be referred to as a *vertex of level* 0. The number of cells that share a vertex (i.e., the number of elements in each equivalence class) is called the *valence* of the vertex.

We make the following assumptions:

(A1) The points in the interior of an edge are in the extended neighbor relation with points in the interior of at most one other edge, and

(A2) the valences of all inner vertices are at least 3.

The assumption (A1) means that at most two edges can be glued by the neighbor relation (in other words, the input control mesh is a manifold). To give $\Diamond \times \mathcal{J}$ a topological structure, we define a subset S as *open* if and only if $S \cap (\Diamond \times \{i\})$ is open for all $i \in \mathcal{J}$ in the natural topology of $\Diamond$. With this definition and under Assumptions (A1) and (A2) described above, the set

$$\mathcal{E}^0 = (\Diamond \times \mathcal{J})/_{\hat{\simeq}} \tag{11}$$

becomes a 2-manifold (possibly) with boundary.

The *boundary* (edges, vertices) consists of all boundary points as described in Section 2.1. A *boundary cell* is a fundamental cell with at least one point on the boundary. All other fundamental cells are called *inner cells*, cf. Figure 1.

If the valence of an inner vertex is different from 4 in case $\Diamond = \square$ and 6 in case $\Diamond = \triangle$, the vertex is called *extraordinary*. All other vertices are said to be *regular*.

In order to keep the number of types of cells small (see below) we make the following assumption:

(A3) All cells have at most one extraordinary vertex. In the case of Butterfly subdivision, we additionally assume that for any given cell, there is at most one extraordinary vertex among the vertices of all the cells that share a vertex with it.

In the case of Catmull-Clark and Loop subdivision, we call a fundamental cell *irregular* if it has an extraordinary vertex, which is then unique by the above assumption. All other fundamental cells are said to be *regular*, see again Figure 1.

In the case of the Butterfly subdivision scheme, the cell that has an extraordinary vertex and their neighboring cells will all be called irregular. More specifically, the cells that contain an extraordinary vertex will be called irregular cells of *type A*, the cells that share an edge with the type A irregular cells are of *type B*, and the cells that share only a vertex with an irregular cell of type A will be referred to as *type C*, see Figure 2, left. All other cells are *regular*.

We call $\mathcal{E}^0$, defined in (11), the *extended domain manifold of level* 0. We obtain the *domain manifold* $\mathcal{M}$ as the closure of the manifold that is obtained by omitting certain cells from $\mathcal{E}^0$. In particular, in the case of Catmull-Clark and Loop subdivision, we obtain the manifold $\mathcal{M}$ by omitting all boundary cells from $\mathcal{E}^0$ (cf. Figure 1, right), whereas in

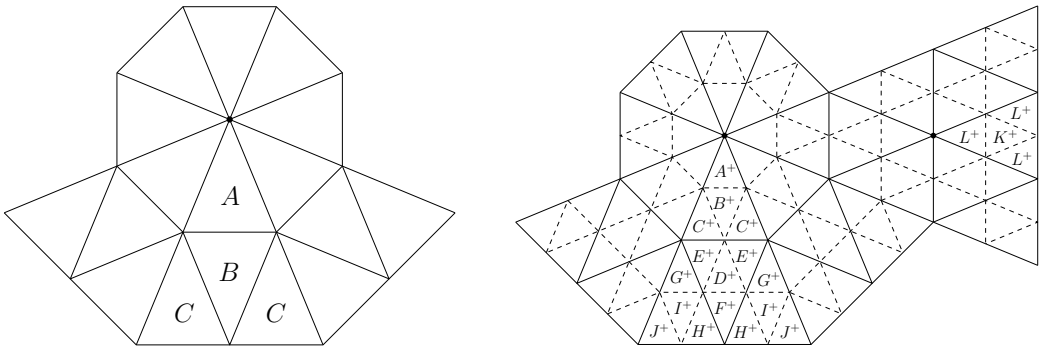


Figure 2: Types of cells of level ℓ (left) and $\ell+1$ (right) around an extraordinary vertex for Butterfly subdivision. Additionally, types of regular cells of level $\ell+1$ (K^+ and L^+) are marked (right). The classification of cells of level $\ell+1$ will be used in Section 5.3.

the case of Butterfly subdivision we obtain it by omitting two layers of cells along the boundary.

Recall that the concept of the extended domain manifold is introduced in order to deal with subdivision splines at boundaries. In the case of subdivision surfaces without boundaries, the extended domain manifold and the domain manifold $\mathcal{M}$ are the same and do not possess boundaries. Otherwise, these manifolds are different and both have boundaries.

We make two additional assumptions:

(A4) All boundary vertices of the extended domain manifold $\mathcal{E}^0$ have valence at most 3 for $\Diamond = \square$ and at most 5 for $\Diamond = \triangle$, and

(A5) all inner vertices of the extended domain manifold $\mathcal{E}^0$ that belong to a cell which is omitted when creating $\mathcal{M}$ are regular.

According to Assumption (A4), there are three types of boundary vertices for the Catmull-Clark scheme, namely convex corners (valence 1), regular boundary vertices (valence 2) and non-convex corners (valence 3). The situation is similar for the Loop and Butterfly subdivision schemes, where we distinguish between two types of convex corners (valence 1 and 2), regular boundary vertices (valence 3) and two types of non-convex corners (valence 4 and 5). Assumptions (A4) and (A5) allow to deal with boundaries simply by using the subdivision rules for regular vertices.

The fundamental cells induce a partition of $\mathcal{E}^0$ into *cells of level* 0. These cells share edges and vertices but have mutually disjoint interiors.

The mother cell $\Diamond$ is subdivided into *mother cells of level* ℓ for $\ell > 0$ by the lines

$$x = j\,2^{-\ell}, y = j\,2^{-\ell}, \text{ and additionally } x + y = j\,2^{-\ell} \text{ in the case } \Diamond = \triangle$$

for $j = 1, \ldots, 2^\ell - 1$. This can be seen as a series of *splitting steps*, where each cell is subdivided into 4 smaller ones. The fundamental cells and the cells of level 0 of the extended domain manifold inherit this subdivision into higher order cells. Consequently, we obtain a partition of $\mathcal{E}^0$ into *cells of level* ℓ. The notions of boundaries, boundary cells, inner cells, and vertices extend naturally to these cells. In particular, we denote the *set of vertices of level* ℓ by V^ℓ.

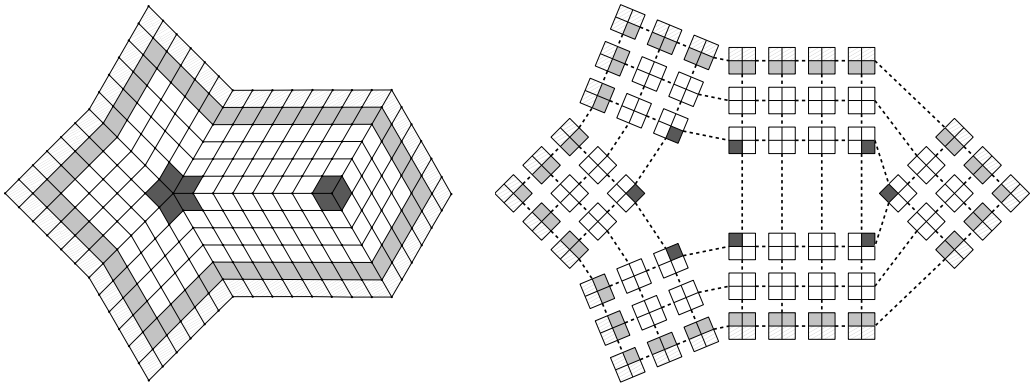


Figure 3: The extended domain manifold of Figure 1 after one splitting step. After discarding one layer of cells along the boundary (dashed) of $\mathcal{E}^0$, the remaining ones are again categorized as boundary (light gray), regular (white), or irregular (dark gray) cells of level 1. Their collection forms the extended domain manifold of level 1. The neighbor relation (right; indicated by dashed lines) remains unchanged.

We now define the *extended domain manifold* $\mathcal{E}^\ell$ *of level* ℓ as the closure of the manifold that is obtained by omitting certain cells of level ℓ in $\mathcal{E}^{\ell-1}$ ($\ell = 1, 2, \ldots$). In particular, in the case of Catmull-Clark and Loop subdivision schemes we omit the boundary cells of level ℓ in $\mathcal{E}^{\ell-1}$, while in the case of Butterfly subdivision we omit again two layers of cells of level ℓ along the boundary. Due to the geometric properties of the subdivision of the elementary cell, this procedure gives the interior of the domain manifold $\mathcal{M}$ for all levels ℓ.

Example 6. Applying one splitting step to the faces of the control mesh and to the corresponding fundamental cells in Example 5 gives the mesh shown in Figure 3. After removing the boundary cells (dashed cells in Figure 3, right), the remaining faces correspond to the cells of level 1 that form the extended domain manifold $\mathcal{E}^1$. Again, these cells are classified as boundary (light gray), regular (white), and irregular (dark gray) ones. The regular and irregular (inner) cells again cover the domain manifold $\mathcal{M}$.

Note that applying a subdivision step does not change the total number of extraordinary vertices. If the cells of level 0 of a given subdivision surface do not satisfy Assumption (A3) we apply one or two splitting steps and re-define the extended domain manifold of level 1 or 2 as $\mathcal{E}^0$.

4.2. Subdivision splines

For each level ℓ, we define the *space* $\mathcal{L}^\ell$ *of globally continuous piecewise (bi-)linear functions* on the extended domain manifold $\mathcal{E}^\ell$ whose restrictions to the cells of level ℓ are bilinear for $\Diamond = \square$ or linear for $\Diamond = \triangle$. Let $\Lambda^\ell = (\lambda_v^\ell)_{v \in V^\ell}$ be the basis of hat functions for $\mathcal{L}^\ell$, including the truncated hat functions at boundary vertices.

Any function $f^\ell \in \mathcal{L}^\ell$ can be expressed as a linear combination of these hat functions. We collect the coefficients in a row vector $c^\ell = (c_v^\ell)_{v \in V^\ell}$ and write f^ℓ as a product of two vectors, $f^\ell = c^\ell \Lambda^\ell$.

Linear subdivision schemes (including Catmull-Clark, Loop and Butterfly) are described by linear operators

$$\mathcal{S}^{\ell+1} : \mathcal{L}^\ell \to \mathcal{L}^{\ell+1} : c^\ell \Lambda^\ell \mapsto \mathcal{S}^{\ell+1}(c^\ell \Lambda^\ell) = (c^\ell R^{\ell+1}) \Lambda^{\ell+1}, \tag{12}$$

which induce linear maps $c^\ell \mapsto c^\ell R^{\ell+1}$ of the coefficient vectors. In particular, applying these operators to the hat functions (by replacing c^ℓ in (12) with the identity matrix[1]) gives

$$\mathcal{S}^{\ell+1}(\Lambda^\ell) = R^{\ell+1}\Lambda^{\ell+1}. \tag{13}$$

The *refinement matrices* $R^{\ell+1}$ are highly sparse, see [32]. A matrix element $R^{\ell+1}_{vw}$ is non-zero only if v and w can be connected by at most two (Catmull-Clark and Loop subdivisoin) or three (Butterfly subdivision) edges.

The refinement matrices are derived from the subdivision rules of Catmull-Clark, Loop and Butterfly schemes. Special rules (which are derived from the rules of B-splines with multiple knots) can be used along the boundaries for the Catmull-Clark scheme if no boundary vertices with valence 3 are present. Alternatively, modified rules, such as those in [6], can be used at boundaries (allowing also for extraordinary boundary points). In the case of Loop subdivision, the rules developed in [18] can be applied. The boundary in the case of the Butterfly scheme can be treated as proposed in [37].

Starting with a function $f^\ell \in \mathcal{L}^\ell$, a *subdivision spline* σ^ℓ of level ℓ is defined as the limit of a sequence of piecewise (bi-)linear functions created by the refinement operators $\mathcal{S}^\ell$,

$$\sigma^\ell = \lim_{L\to\infty} \mathcal{S}^L \cdots \mathcal{S}^{\ell+1} f^\ell. \tag{14}$$

In each refinement step, the domain of the function is modified by removing cells along the boundary as described above. The domain of the limit function is the domain manifold $\mathcal{M}$.

The existence of the limit is guaranteed for the operators obtained from Catmull-Clark, Loop and the Butterfly subdivision schemes. The existence of limit functions for more general schemes has been analyzed in the rich literature on subdivision, see e.g. [25].

In particular, applying (14) to the hat functions λ^ℓ_v gives *subdivision blending functions*

$$\gamma^\ell_v = \lim_{L\to\infty} \mathcal{S}^L \cdots \mathcal{S}^{\ell+1} \lambda^\ell_v$$

of level ℓ. Similarly to the hat functions, we collect them in column vectors

$$G^\ell = (\gamma^\ell_v)_{v\in V^\ell} = \lim_{L\to\infty} \mathcal{S}^L \cdots \mathcal{S}^{\ell+1} \Lambda^\ell.$$

Due to the linearity of the refinement operators $\mathcal{S}^\ell$ we can express a subdivision spline (14) of level ℓ as a linear combination $\sigma^\ell = c^\ell G^\ell$ since

$$\sigma^\ell = \lim_{L\to\infty} \mathcal{S}^L \cdots \mathcal{S}^{\ell+1} c^\ell \Lambda^\ell = c^\ell \lim_{L\to\infty} \mathcal{S}^L \cdots \mathcal{S}^{\ell+1} \Lambda^\ell = c^\ell G^\ell.$$

In addition, using (13) and the linearity of the refinement operators implies the refinement equation

$$\begin{aligned} G^\ell &= \lim_{L\to\infty} \mathcal{S}^L \cdots \mathcal{S}^{\ell+1}(\Lambda^\ell) = \lim_{L\to\infty} \mathcal{S}^L \cdots \mathcal{S}^{\ell+2}(R^{\ell+1}\Lambda^{\ell+1}) = \\ &= R^{\ell+1} \lim_{L\to\infty} \mathcal{S}^L \cdots \mathcal{S}^{\ell+2}(\Lambda^{\ell+1}) = R^{\ell+1} G^{\ell+1}, \end{aligned} \tag{15}$$

[1]When replacing a vector c^ℓ by a matrix, the operator $\mathcal{S}^{\ell+1}$ is applied to a vector and therefore to each component separately.

see also (1). The subdivision blending functions for all three considered schemes are normalized, which corresponds to the property of affine invariance. In addition, the Catmull-Clark and Loop schemes produce non-negative blending functions, thereby ensuring the convex-hull property.

In the case of Catmull-Clark and Loop subdivision, the *support* of each subdivision blending function γ_v^ℓ is the interior of the 2-ring neighborhood of the vertex $v \in V^\ell$ with respect to the inner cells of level ℓ, due to the sparsity properties of the refinement matrices. This neighborhood consists of the cells that contain v and their neighboring cells. Note that the 2-ring neighborhood consists of different numbers of cells for different types of vertices, see Example 7.

In the case of Butterfly subdivision, the support of each subdivision blending function is *contained* in the interior of the 3-ring neighborhood of cells of the vertex. Again, the number of cells in this neighborhood depends on the types of the vertices within. We say that the support is *regular* if the interior of the 2-ring (for Catmull-Clark and Loop subdivision) or 3-ring (for Butterfly subdivision) neighborhood contains no extraordinary vertices of level ℓ, and *non-regular* otherwise.

Example 7. The supports of selected Catmull-Clark subdivision basis functions, defined with respect to the extended domain manifold $\mathcal{E}^1$ in Figure 3, are shown in Figure 4. Notice that the supports of the functions that correspond to vertices near the boundary are truncated. The vertices that correspond to functions with regular supports are marked by a circle $\circ$, whereas a bullet $\bullet$ identifies functions with non-regular supports.

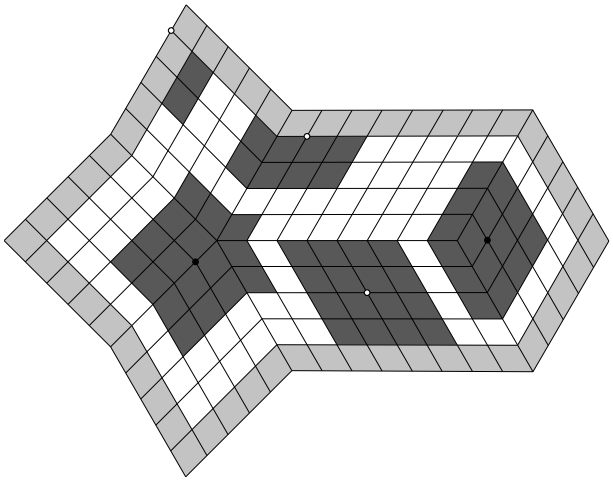


Figure 4: Supports (dark gray) of selected Catmull-Clark subdivision blending functions of level 1 in Example 7, cf. Figure 3. The boundary faces (light gray) correspond to cells that are not contained in the domain manifold $\mathcal{M}$. Circles and bullets mark vertices that correspond to subdivision blending functions with regular and non-regular supports, respectively.

Example 8. Figure 5 shows an example of a subdivision blending function for Catmull-Clark (left) and Butterfly (right) subdivision schemes. The functions correspond to extraordinary vertices of valence 5 and 7, respectively. The graphs of the blending functions are plotted above the 2- and 3-ring neighborhoods of cells in $\mathcal{M}$, surrounding the extraordinary vertex, that were embedded into $\mathbb{R}^2$ (see Figure 5). The embeddings, shown in the figure, are translated in the z-direction away from the graphs to improve clarity.

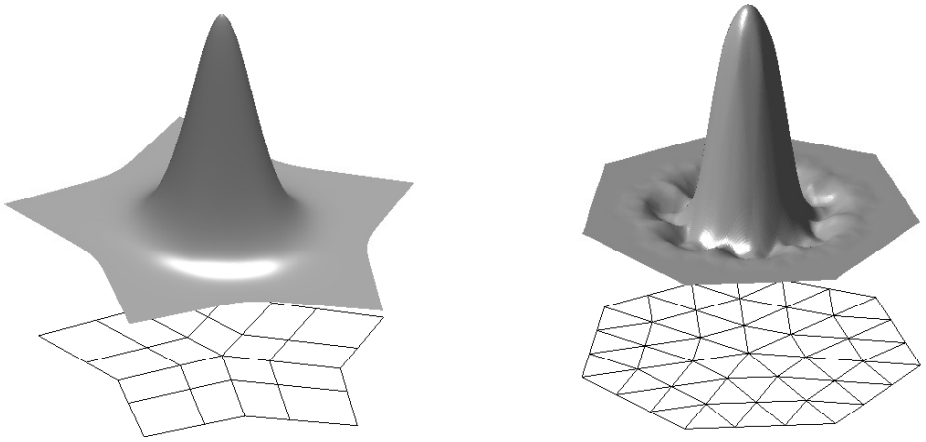
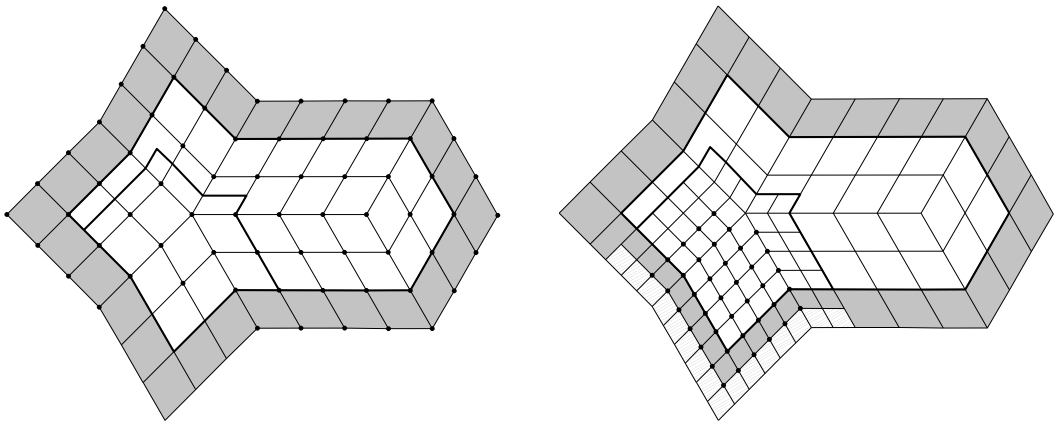


Figure 5: Graphs of subdivision blending functions for Catmull-Clark (left) and Butterfly (right) subdivision schemes, corresponding to extraordinary vertices of valence 5 and 7, respectively. The 2- and 3-ring neighborhoods of cells in $\mathcal{M}$, surrounding the extraordinary vertex, are embedded into $\mathbb{R}^2$.

The construction of a hierarchical generating system, which was summarized in Section 2.2, can now be applied to subdivision splines simply by considering the generating systems of subdivision blending functions. This leads to *hierarchical subdivision splines*. An example in the Catmull-Clark case with two levels is shown in Figure 6.



(a) (b)

Figure 6: Hierarchical Catmull-Clark subdivision blending functions of level 0 (left) and level 1 (right). Bullets mark selected blending functions according to the domain hierarchy with two levels. The subdomains $\mathcal{M}^0$ and $\mathcal{M}^1$ are outlined with thick lines.

In addition to the domain manifold $\mathcal{M}$, this construction is based on a hierarchy of subdomains $\mathcal{M}^\ell$ which specify the regions that are selected for adaptive refinement. These subdomains have to be chosen in a suitable way (especially in the context of isogeometric analysis), in order to guarantee the linear independence of the resulting hierarchical generating system, cf. Theorem 3. These constraints are discussed in the next section.

We restrict our discussion to subdomains which are unions of cells of a certain level. More precisely, each subdomain $\mathcal{M}^\ell$ is assumed to be the union of cells of level ℓ or even

of level $\ell - 1$ for $\ell > 0$. These two cases were considered already in [33], where they were referred to as the *weak* and the *strong condition* on the subdomain hierarchy, respectively. The recent article [35], which focuses on Catmull-Clark subdivision, uses a construction that relies on the strong condition.

In order to construct the hierarchical generating system, we need to decide whether the support of a subdivision blending function γ_v^ℓ is contained in the subdomains $\mathcal{M}^\ell$ and $\mathcal{M}^{\ell+1}$ or not. These decisions are relatively simple for Catmull-Clark and Loop subdivision since the supports are simply the interiors of unions of cells. However, the situation is more subtle for Butterfly subdivision, where the supports have a fractal nature. Nevertheless, since we only consider restricted domain hierarchies, it suffices to analyze whether a cell of level ℓ or $\ell + 1$ contributes to the support of a given blending function γ_v^ℓ or not in the case of the strong or the weak condition, respectively, see Figure 7.

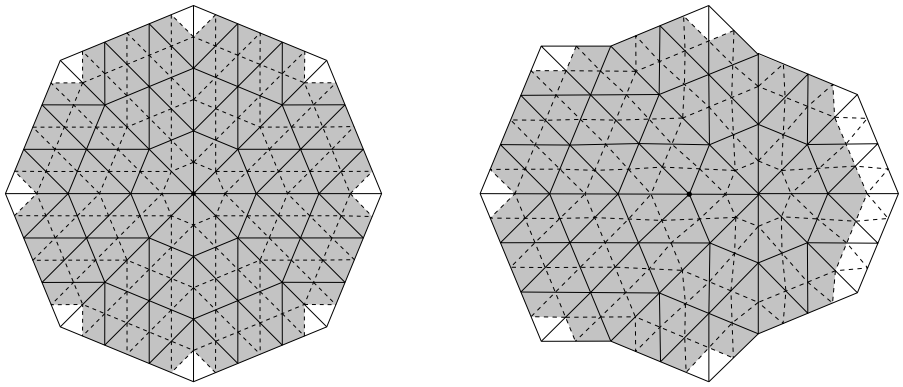


Figure 7: Cells of level $\ell + 1$ (indicated by dashed lines) that contribute to the support of a Butterfly blending function of level ℓ. Note that the support of the blending function shown on the right contains an extraordinary vertex, which is not the central vertex.

Finally we note that the construction extends to *truncated* hierarchical generating systems for subdivision splines, since the domain hierarchies satisfy condition (4). This is obvious for Catmull-Clark and Loop subdivision, since the blending functions are non-negative. For subdomain hierarchies that satisfy the weak condition (see above), a tedious but straightforward analysis confirms this fact in the case of Butterfly subdivision also.

5. Safe subdomains for subdivision splines

Global linear independence of subdivision blending functions on manifolds without boundaries is well understood: For the case of Catmull-Clark and Loop subdivision, it has been shown that these schemes generate blending functions that are linearly independent on all domain manifolds $\mathcal{M}$ except for pathological cases [24], which are ruled out by Assumption (A3). The blending functions of the Butterfly scheme are linearly independent due to their interpolatory nature.

In this section we discuss the linear independence of Catmull-Clark, Loop and Butterfly hierarchical subdivision blending functions *on subdomains*. In particular, we focus on subdomains that are cells of level ℓ or level $\ell+1$, as required when using domain hierarchies satisfying the weak or the strong condition.

We describe a general procedure to obtain a catalog of safe subdomains, which is required for ensuring linear independence of the hierarchical generating system when con-

structing the domain hierarchy as described in Section 3. We provide such catalogs for Catmull-Clark, Loop and Butterfly subdivision.

Linear independence of Catmull-Clark and Loop subdivision blending functions on particular subdomains was already studied in [24], using eigenanalysis of the refinement matrices. This produced results that are equivalent to the findings reported below. However, the relation to linear independence of hierarchical generating systems was not noted.

We focus on the linear independence of hierarchical generating systems and present the results in the form of catalogs, making them directly usable for adaptive mesh refinement. The cases of Catmull-Clark and Loop subdivision appear as simple – though clearly important – special cases. Indeed, the same procedure allows us to establish similar results for any subdivision scheme that can be defined as described in Section 4.2, and we demonstrate this for Butterfly subdivision. Also, it should be noted that we certify linear independence without performing an eigenanalysis of the refinement matrices.

5.1. Finding safe subdomains

We first present a method to analyze whether a chosen subdomain S is G^ℓ-safe or not. We consider the subdivision blending functions of level $k \geq \ell$ whose supports possess a non-empty intersection with the subdomain S. These functions form a subvector G^k_S of G^k. Any subdivision spline σ^k of level k has a representation

$$\sigma^k|_S = c_S G^k_S|_S$$

with a coefficient row vector c_S. First, we consider the case $\ell = k$. According to Definition 1, the subdomain S is G^k-safe if and only if

$$0 = c_S G^k_S|_S \;\Leftrightarrow\; c_S = \mathbf{0}, \tag{16}$$

where $\mathbf{0}$ is a row null vector of dimension $\dim G^k_S$. For subdivision splines of level ℓ with $\ell < k$, we use the refinement equation to represent the function $\sigma^\ell|_S$ with respect to G^k_S,

$$\sigma^\ell|_S = c_S G^\ell_S|_S = c_S R^{\ell+1}_S \cdots R^k_S G^k_S,$$

where R^{j+1}_S is the sub-matrix of R^{j+1} containing only the rows and columns that correspond to the elements of the sub-vectors G^j_S and G^{j+1}_S, respectively. The following lemma gives a sufficient condition for G^ℓ-safety of a chosen subdomain S. It should be noted that the same argument has been used also in the proofs in [24]. We formulate it here as a lemma since we will to refer to it later.

Lemma 9. *A G^k-safe subdomain S is G^ℓ-safe, where $\ell \leq k$, if the matrix $R^{\ell+1}_S \cdots R^k_S$ has full row rank.*

Proof. Since S is G^k-safe, we use the observation (16) to conclude that

$$0 = c_S G^\ell_S|_S = c_S R^{\ell+1}_S \cdots R^k_S G^k_S|_S \tag{17}$$

is equivalent to

$$c_S R^{\ell+1}_S \cdots R^k_S = \mathbf{0}, \tag{18}$$

where $\mathbf{0}$ is a row null vector of dimension $\dim G^k_S$. If the matrix $R^{\ell+1}_S \cdots R^k_S$ has full row rank, then this is equivalent to $c_S = \mathbf{0}$, where this right-hand side is a row null vector of dimension $\dim G^\ell_S$. Thus, the functions $G^\ell_S|_S$ are linearly independent. □

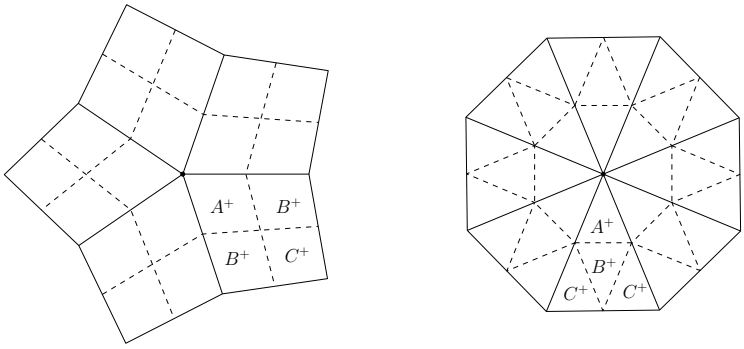


Figure 8: An irregular cell is subdivided into regular cells B^+ and C^+ (three in total), and one irregular cell of type A^+, for Catmull-Clark (left) and Loop (right) subdivision schemes, respectively.

On the other hand, if the matrix $R_S^{\ell+1} \cdots R_S^k$ does *not* have full row rank, the subdomain S is not G^ℓ-safe, independently of whether it is G^k-safe for some $k > \ell$ or not. This is made precise in the following lemma.

Lemma 10. *The subdomain S is not G^ℓ-safe if the matrix $R_S^{\ell+1} \cdots R_S^k$, $\ell \leq k$, does not have full row rank.*

Proof. If the matrix $R_S^{\ell+1} \cdots R_S^k$ does not have full row rank, then there exists a non-zero vector c_S such that (18) holds. Based on (17), we conclude that functions G_S^ℓ are linearly dependent on the subdomain S, thus S is not G^ℓ-safe. □

5.2. Catmull-Clark subdivision

Applying one subdivision step splits a cell of level ℓ into four cells of level $\ell+1$. For an irregular cell, we classify the cells of level $\ell+1$ obtained by splitting it into three different types A^+, B^+ and C^+ according to their location with respect to the extraordinary vertex, see Figure 8, left. Consequently, we study the linear independence of G^ℓ with respect to six different types of cells, namely regular and irregular cells of level ℓ, cells of level $\ell+1$ obtained when splitting a regular cell, and the three types A^+, B^+ and C^+.

The results concerning linear independence on these six types of cells are summarized in Table 1, left. Almost all of them are already covered by results in [24]. We obtained them as special cases of our general procedure and present them here in the form of a catalog of safe subdomains, thereby making them directly usable for adaptive mesh refinement of hierarchical subdivision splines, see Section 3.2. It should be noted that the construction described in [35] uses the safe subdomains of level ℓ only (i.e., the second column of the table).

We briefly discuss how the results in the table can be proved in our framework. The row for valence 4 (shown in gray) contains the results obtained for a regular cell of level ℓ and for cells of level $\ell+1$ obtained when splitting it. The classification of the cells into types A^+, B^+ and C^+ does not apply in this case. There are 16 blending functions whose supports intersect these cells, which are then equal to the polynomial segments of the 16 uniform bicubic tensor-product B-splines on a regular tensor-product grid. These functions are linearly independent on any open subset of that cell due to the local linear independence of tensor-product B-splines (see e.g. [24, Lemma 5.1]). Consequently, all these cells are G^ℓ-safe.

ν	ℓ	$\ell+1$		
		A^+	B^+	C^+
3	✓	×	✓	✓
4	✓		✓	
5	✓	✓	×	×
6	✓	✓	×	×
7	✓	✓	×	×
8	✓	✓	×	×
9	✓	✓	×	×
10	✓	✓	×	×
11	✓	✓	×	×
12	✓	✓	×	×

ν	ℓ	$\ell+1$		
		A^+	B^+	C^+
3	✓	✓	✓	✓
4	✓	✓	✓	✓
5	✓	✓	✓	✓
6	✓		✓	
7	✓	✓	×	×
8	✓	✓	×	×
9	✓	✓	×	×
10	✓	✓	×	×
11	✓	✓	×	×
12	✓	✓	×	×

Table 1: G^ℓ-safety of cells of level ℓ and $\ell+1$ (types A^+, B^+, C^+) for Catmull-Clark (left) and Loop (right) subdivision schemes for different valences ν. The check marks and crosses indicate whether the cells are G^ℓ-safe or not, respectively.

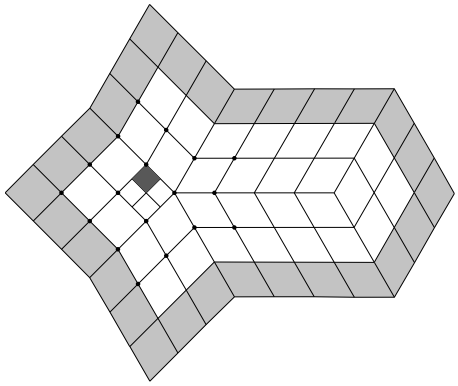


Figure 9: A set of 18 Catmull-Clark subdivision blending functions of level ℓ (the corresponding vertices are marked by bullets) whose restrictions to the cell of type B^+ (shown in dark gray) for valence 5 are linearly dependent.

In particular, this implies that *any set of regular cells of level k is G^k-safe*. We proved the remaining checkmarks in the table by combining this fact with Lemma 9. More precisely, we considered the subset S of regular cells of some sufficiently high level $k > \ell$ that are contained in the cell under consideration. We used symbolic computation to confirm that the matrix in the lemma possesses full row rank. Lemma 2 then allows us to extend the result to the entire cell under consideration. For the first two columns of the table, the choice of k is related to [24, Conjecture 2].

Next, we consider the 16 crosses in the lower right part of the left table. If an irregular cell of level ℓ contains an extraordinary vertex of valence other than 3, the Catmull-Clark subdivision blending functions of level ℓ are not linearly independent on the cells of type B^+ and C^+. We recall the proof of this fact by a counting argument from [24]: The number of subdivision blending functions whose supports contain the considered cell is $2\nu+8$, where ν is the valence of the extraordinary vertex. For $\nu \geq 5$ this number exceeds 16, which is the dimension of the space of bicubic polynomials, see Figure 9. This observation can also be justified by using Lemma 10 with $k = \ell+1$, since the matrix has more rows than columns in this case. Finally, we use Lemma 10 with $k = \ell+1$ to prove the linear dependency of subdivision blending functions on the cells of type A^+, which has been visualized already in [24, Fig. 4.4].

5.3. Loop subdivision

Similar to the previous section, we study linear independence of G^ℓ with respect to six different types of cells. More precisely we consider regular and irregular cells of level ℓ, cells of level $\ell+1$ obtained when splitting a regular cell, and the three types A^+, B^+ and C^+ obtained when splitting an irregular cell, see Figure 8, right. The results concerning linear independence on these six types of cells are summarized in Table 1, right. All entries in the table are covered by results in [24] and we sketch how they can be confirmed by using our framework.

The row for valence 6 (shown in gray) contains the results obtained for a regular cell of level ℓ and for cells of level $\ell+1$ obtained when splitting it. The classification of the cells into types A^+, B^+ and C^+ does not apply in this case. The 12 blending functions whose supports intersect these cells are known to be equal to the polynomial segments of the C^2 box splines of total degree 4 on a type-I triangulation. The local linear independence of box splines on type-I triangulations implies G^ℓ-safety of the regular cells of all levels.

In particular, this implies, as in the case of Catmull-Clark subdivision, that *any set of regular cells of level k is G^k-safe*. The remaining entries of the table are then derived as in the previous section, using Lemmas 9 and 2 for the checkmarks, where the set S is the set of regular cells of some level $k > \ell$ contained in the cell under consideration, and Lemma 10 otherwise.

5.4. Butterfly subdivision

Recall that there are three types A, B and C of irregular cells of level ℓ, see Figure 2, left. Once again, when analyzing the G^ℓ-safety of a cell of level $\ell+1$, we need to distinguish between different types of cells obtained by subdividing the irregular cells. We arrive at 10 different types A^+ to J^+ of cells of level $\ell+1$, see Figure 2, right. In addition, we also need to distinguish between two different types of cells obtained by subdividing the regular cells, which we will denote by K^+ and L^+, see again Figure 2, right. In summary, we need to analyze G^ℓ-safety on regular cells of level ℓ, on the three types of irregular cells of level ℓ, and on the 12 types A^+ to L^+.

Even in the vicinity of regular vertices, the blending functions generated by the Butterfly subdivision scheme are not piecewise polynomials. We are not aware of any results regarding local linear independence. The next result follows immediately from the interpolatory nature of Butterfly subdivision.

Lemma 11. *Any subset of the vertex set V^ℓ that is contained in the domain manifold $\mathcal{M}$ is G^ℓ-safe for Butterfly subdivision.*

We combine this observation with Lemma 9 and the second part of Lemma 2 to prove that some of the cells are G^ℓ-safe. More precisely, for a suitable $k > \ell$, we choose the set S to be the subset of the vertex set V^k that is contained in the cell under consideration. We then analyze the row rank of the corresponding matrix in Lemma 9, whose entries are simply the values of the subdivision blending functions of level ℓ at the vertices forming S. If this matrix possesses full row rank, then the second part of Lemma 2 allows us to conclude that the entire cell is G^ℓ-safe.

In addition, we rely on Lemma 10 for identifying cells that are not G^ℓ-safe, simply by analyzing the rank of the matrix specified there. In this case, the set S is the entire cell under consideration. The matrix entries are then the values of the subdivision blending

	ℓ			$\ell+1$									
ν	A	B	C	A^+	B^+	C^+	D^+	E^+	F^+	G^+	H^+	I^+	J^+
3	✓	✓	✓	×	✓	✓	✓	✓	✓	✓	✓	✓	✓
4	✓	✓	✓	×	×	✓	✓	✓	✓	✓	✓	✓	✓
5	✓	✓	✓	×	×	✓	✓	✓	✓	✓	✓	✓	✓
6		✓						K^+: ✓	L^+: ✓				
7	✓	✓	✓	×	×	✓	✓	✓	✓	✓	✓	×	×
8	✓	✓	×	×	×	✓	✓	✓	×	×	×	×	×
9	✓	✓	×	×	×	×	×	✓	×	×	×	×	×
10	✓	✓	×	×	×	×	×	×	×	×	×	×	×
11	✓	×	×	×	×	×	×	×	×	×	×	×	×
12	✓	×	×	×	×	×	×	×	×	×	×	×	×

Table 2: G^ℓ-safety of cells of level ℓ (types A, B, C) and level $\ell+1$ (types A^+ to L^+) for Butterfly subdivision for different valences ν. The check marks and crosses indicate whether the cells are G^ℓ-safe or not, respectively.

functions at vertices from the vertex set V^k for a suitable $k > \ell$ within the cell and located on the two layers (with respect to the grid of level k) surrounding it.

Our results are summarized in Table 2. The results for valences other than 7, 9 and 11 were obtained using symbolic computations. The remaining high odd valencies were obtained by numerical computations in MATLAB as the coefficients involved [37] for these valences do not admit representations by radicals. This complicates the symbolic manipulation of the resulting expressions considerably.

5.5. Using the catalogs for maintaining linear independence

We use the catalogs presented in Tables 1 (right) and 2 for Loop and Butterfly subdivision, respectively, in order to perform adaptive refinement with the help of the simple refinement algorithm that was presented in Section 3.2. The different catalogs lead to different results, as shown in the following example.

Example 12. We consider a subdomain hierarchy with only one level $\mathcal{M} = \mathcal{M}^0 \supseteq \mathcal{M}^1 = \emptyset$, where $\mathcal{M}$ is a domain manifold consisting of triangular cells, see Fig. 10. After selecting a region Z (dark gray) for refinement, the algorithm generates a new subdomain hierarchy

$$\mathcal{M} = \tilde{\mathcal{M}}^0 \ \supseteq \ \tilde{\mathcal{M}}^1 = \mathcal{M} \setminus W_Z \ \supseteq \ \tilde{\mathcal{M}}^2 = \emptyset,$$

where the subset W_Z is defined as

$$W_Z = [U]_0 \cap (\mathcal{M} \setminus Z), \quad U = \bigcup_{\mathcal{M} \setminus Z \supseteq T \in C^0} T,$$

cf. Eq. (7). In order to construct $\tilde{\mathcal{M}}^1$, we first generate U (shown in white) as the union of all cells from the catalog C^0 (see Tables 1 and 2) that are outside Z. In the second step, we identify all blending functions (which correspond to the vertices that have been marked by circles) of level 0 whose supports do not overlap with U. For some of them (hollow circles), the support contains cells (shown in black) that are not contained in Z. Adding these cells to Z gives $\tilde{\mathcal{M}}^1$ (black and dark gray). In the case of Loop subdivision (Fig. 10, left), we need to add 8 cells of level 1 to obtain $\tilde{\mathcal{M}}^1$. For Butterfly subdivision (Fig. 10, right), we need to add 6 cells of level 1. Although some cells (light gray) are not contained in U, they are not included in $\tilde{\mathcal{M}}^1$ since each Butterfly blending function acting on them takes non-zero values on U as well and hence the second part of Lemma 2 applies.

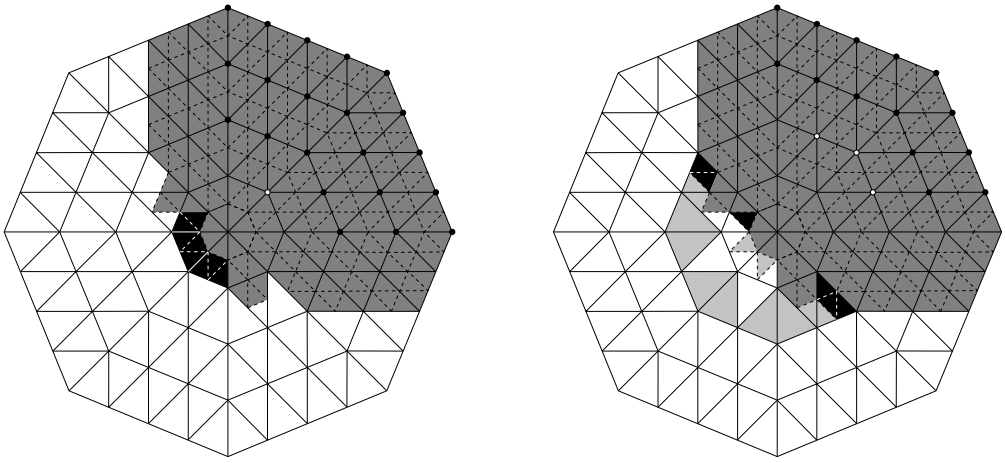


Figure 10: The result of the algorithm that maintains linear independence of (truncated) hierarchical generating system created by adaptive refinement for Loop (left) and Butterfly (right) subdivision. A region is marked for refinement (dark gray) and is then enlarged by adding certain cells (black) to ensure linear independence. In the case of Butterfly subdivision, some cells (light gray), which are not G^0–safe individually, do not need to be refined since their union with the cells shown in white belongs to C^0.

6. Conclusion

We have extended the construction of (truncated) hierarchical generating systems, which was presented in [36] for functions defined on domains in $\mathbb{R}^2$, to functions defined on domain manifolds. In order to guarantee linear independence, the framework of safe subdomains has been introduced.

Based on these abstract results, we studied subdivision splines generated by the Catmull-Clark, Loop and Butterfly subdivision schemes. In particular, we provided catalogs of safe subdomains that allow the design of domain hierarchies with linearly independent (truncated) hierarchical generating systems. For the two approximating schemes, these catalogs are almost entirely covered by the results in [24], which we summarized and re-confirmed using our framework. In addition we were able to obtain a similar catalog also for the modified Butterfly subdivision introduced in [37]. It should be noted that the resulting conditions for linearly independent (truncated) hierarchical generating systems based on the Butterfly scheme are far more restrictive than those for Catmull-Clark and Loop subdivision, due to the larger number of cells that are unsafe.

Potential applications include adaptive subdivision surface fitting and numerical simulation using isogeometric analysis. Truncated hierarchical Catmull-Clark subdivision is explored in the recent article [35]. Clearly, besides using Catmull-Clark subdivision [1, 20], which operates on quadrangular meshes, subdivision schemes for triangular meshes (such as Loop subdivision) are of great interest for analysis as well, cf. [2]. Moreover, we feel that the use of *interpolatory* subdivision schemes in isogeometric analysis might be appealing for practitioners in the finite element community due to the simplicity of enforcing Dirichlet boundary conditions.

In order to keep the presentation simple, our results have been formulated for three specific subdivision schemes, which are also of substantial practical interest. It is possible to use our approach for other schemes, in particular for any linear scheme that provides a non-trivial system of safe subdomains.

Acknowledgments. The authors wish to thank Malcolm Sabin for his support, encouragement, and helpful suggestions concerning this work, and Pieter Barendrecht for making his implementation of subdivision splines available, which was a useful starting point for us. We also thank Neil Dodgson for his support during the first author's visit to the Computer Laboratory at the University of Cambridge.

The first two authors were supported by the European Union through FP7 Grant ITN INSIST, GA no. 289361. The last author was supported by the Engineering and Physical Sciences Research Council through Grants EP/H030115/1 and EP/K503757/1.

References

[1] D. Burkhart, B. Hamann, and G. Umlauf. Iso-geometric analysis based on Catmull-Clark solid subdivision. *Computer Graphics Forum*, 29(5):1575–1784, 2010.

[2] F. Cirak and Q. Long. Subdivision shells with exact boundary control and non-manifold geometry. *Int. J. Numer. Meth. Engrg.*, 88(9):897–923, 2011.

[3] F. Cirak, M. Ortiz, and P. Schröder. Subdivision surfaces: a new paradigm for thin-shell finite-element analysis. *Int. J. Numer. Meth. in Engrg.*, 47(12):2039–2072, 2000.

[4] J.A. Cottrell, T.J.R. Hughes, and Y. Bazilevs. *Isogeometric Analysis: Toward Integration of CAD and FEA*. John Wiley & Sons, 2009.

[5] C. de Boor, K. Höllig, and S. Riemenschneider. *Box splines*. Springer–Verlag, 1993.

[6] T. DeRose, M. Kass, and T. Truong. Subdivision surfaces in character animation. In *Proc. Siggraph 1998*, pages 85–94, New York, NY, USA, 1998. ACM.

[7] N. Dyn, D. Levin, and J. A. Gregory. A butterfly subdivision scheme for surface interpolation with tension control. *ACM Transactions on Graphics*, 9:160–169, 1990.

[8] D. R. Forsey and R. H. Bartels. Hierarchical B-spline refinement. *Computer Graphics*, 22:205–212, 1988.

[9] C. Giannelli, B. Jüttler, and H. Speleers. THB–splines: the truncated basis for hierarchical splines. *Comput. Aided Geom. Design*, 29:485–498, 2012.

[10] C. Giannelli, B. Jüttler, and H. Speleers. Strongly stable bases for adaptively refined multilevel spline spaces. *Adv. Comput. Math.*, 40(2):459–490, 2014.

[11] E. Grinspun, P. Krysl, and P. Schröder. CHARMS: A Simple Framework for Adaptive Simulation. *ACM Trans. Graphics*, 21(3):281–290, 2002.

[12] K. A. Johannessen, T. Kvamsdal, and T. Dokken. Isogeometric analysis using LR B-splines. *Comput. Meth. Appl. Mech. Engrg.*, 269:471–514, 2013.

[13] G. Kiss, C. Giannelli, and B. Jüttler. Algorithms and data structures for truncated hierarchical B-splines. In M. Floater et al., editors, *Mathematical Methods for Curves and Surfaces*, volume 8177 of *Lecture Notes in Computer Science*, pages 304–323. Springer, 2014.

[14] S. Kleiss, C. Pechstein, B. Jüttler, and S. Tomar. IETI – Isogeometric tearing and interconnecting. *Comput. Meth. Appl. Mech. Engrg.*, 247-248:201–215, 2012.

[15] R. Kraft. *Adaptive und linear unabhängige Multilevel B–Splines und ihre Anwendungen*. PhD thesis, Universität Stuttgart, 1998.

[16] X. Li, J. Deng, and F. Chen. Polynomial splines over general T-meshes. *Visual Comput.*, 26:277–286, 2010.

[17] X. Li, J. Zheng, T. W. Sederberg, T.J.R. Hughes, and M.A. Scott. On linear independence of T–spline blending functions. *Comput. Aided Geom. Design*, 29:63–76, 2012.

[18] R. Ling, W. Wang, and D. Yan. Fitting sharp features with Loop subdivision surfaces. *Computer Graphics Forum*, 27(5):1383–1391, 2008.

[19] M. Lounsbery, T. D. DeRose, and J. Warren. Multiresolution analysis for surfaces of arbitrary topological type. *ACM Trans. Graph.*, 16(1):34–73, January 1997.

[20] T. Nguyen, K. Karciauskas, and J. Peters. A comparative study of several classical, discrete differential and isogeometric methods for solving Poisson's equation on the disk. *Axioms*, 3(2):280–299, 2014.

[21] P. Novara and L. Romani. On extraordinary rules of quad-based interpolatory subdivision schemes. *Comput. Aided Geom. Design*, 35–36:225–242, 2015.

[22] P. Novara, L. Romani, and J. Yoon. Improving smoothness and accuracy of modified Butterfly subdivision scheme. *Appl. Math. Comp.*, 2015. in press.

[23] J. Peters and U. Reif. *Subdivision Surfaces*, volume 3 of *Geometry and Computing*. Springer, 2008.

[24] J. Peters and X. Wu. On the local linear independence of generalized subdivision functions. *SIAM J. Numer. Analysis*, 44(6):2389–2407, 2006.

[25] M. Sabin. Subdivision surfaces. In G. Farin, J. Hoschek, and M.-S. Kim, editors, *Handbook of Computer Aided Geometric Design*, chapter 12, pages 309–325. Elsevier, 2002.

[26] R. Schmidt, R. Wüchner, and K.-U. Bletzinger. Isogeometric analysis of trimmed NURBS geometries. *Comput. Meth. Appl. Mech. Engrg.*, 241–244(0):93–111, 2012.

[27] M.A. Scott, M.J. Borden, C.V. Verhoosel, T.W. Sederberg, and T.J.R. Hughes. Isogeometric finite element data structures based on Bézier extraction of T-splines. *Int. J. Numer. Meth. Engrg.*, 88:126–156, 2011.

[28] M.A. Scott, D.C. Thomas, and E.J. Evans. Isogeometric spline forests. *Comput. Meth. Appl. Mech. Engrg.*, 269:222–264, 2014.

[29] T. W. Sederberg, D. L. Cardon, G. T. Finnigan, N. S. North, J. Zheng, and T. Lyche. T-spline simplification and local refinement. *ACM Trans. Graphics*, 23:276–283, 2004.

[30] J. Shen, J. Kosinka, M.A. Sabin, and N.A. Dodgson. Conversion of trimmed NURBS surfaces to Catmull-Clark subdivision surfaces. *Comput. Aided Geom. Design*, 31(7–8), 2014.

[31] H. Speleers and C. Manni. Effortless quasi-interpolation in hierarchical spaces. Technical Report TW647, Dept. of Computer Science, KU Leuven, 2014.

[32] E.J. Stollnitz, A.D. DeRose, and D.H. Salesin. *Wavelets for Computer Graphics*. Morgan Kaufmann, 1996.

[33] A.-V. Vuong, C. Giannelli, B. Jüttler, and B. Simeon. A hierarchical approach to adaptive local refinement in isogeometric analysis. *Comput. Meth. Appl. Mech. Engrg.*, 200:3554–3567, 2011.

[34] X. Wei, Y. Zhang, T.J.R. Hughes, and M.A. Scott. Extended truncated hierarchical Catmull-Clark subdivision. Technical Report 15, ICES, UT Austin, 2015.

[35] X. Wei, Y. Zhang, T.J.R. Hughes, and M.A. Scott. Truncated hierarchical Catmull-Clark subdivision with local refinement. *Comput. Meth. Appl. Mech. Engrg.*, 291:1–20, 2015.

[36] U. Zore and B. Jüttler. Adaptively refined multilevel spline spaces from generating systems. *Comput. Aided Geom. Design*, 31:545–566, 2014.

[37] D. Zorin, P. Schröder, and W. Sweldens. Interpolating subdivision for meshes with arbitrary topology. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 189–192. ACM, 1996.

[38] P. B. Zwart. Multivariate splines with nondegenerate partitions. *SIAM J. Numer. Analysis*, 10(4):665–673, 1973.