Symbolic Spline Computations

Patrick Clarke   Simon Foucart

June 14, 2013

Abstract

This paper introduces a new computational method in spline theory. It is particularly useful, among other features, to determine dimensions of spline spaces. The method is based on a conversion to commutative algebra and allows efficient Hilbert series calculations. It is not restricted to the bivariate and trivariate cases, to simplicial partitions, nor to standard smoothness conditions. The SAGE implementation provides a means to take a fresh look at Rose's freeness conjecture, the swift verification of dimension formulas previously postulated, and allows us to propose dimension formulas for triangulations with hanging vertices.

1 Introduction

This paper introduces a new method based on commutative algebra for computing splines satisfying smoothness conditions. There are almost no restrictions on the domain, on the way it is subdivided, nor on the imposed smoothness conditions. Furthermore, we find that many interesting questions about splines convert to classical and well-understood problems in commutative algebra. For example, determining the dimension formulas of spline spaces with prescribed smoothness and maximal total degree can be cleanly answered in terms of the Hilbert series of a certain ideal. Additionally, once the problem has been translated into commutative algebra, many existing computational commutative algebra software packages can be exploited. We present here implementations of our method and several applications written in SAGE [21].

In Section 2, we begin by reviewing the definition of a spline function over a domain with respect to a subdivision. We also clarify what "smoothness condition" means in terms of the well-definedness of linear differential operators. The ideas are familiar, but making them precise allows us to proceed carefully in our proofs. Furthermore, distinguishing between the spline and the function it defines allows us to uniformly treat situations when the Zariski closure of an element of the subdivision is not $\mathbb{R}^n$. This is a desirable property when thinking about the deformation theory of the subdivison.

With these definitions in place, Section 3 takes the first step towards the translation into commutative algebra with Theorem 3.1. The idea is to convert questions about smoothness into "ideal-difference" conditions. This theorem states that, given some fixed smoothness conditions, there exist ideals $\mathcal{J}_{jk}$ such that a spline satisfies the smoothness conditions if and only if the difference of its values on the j^{th} and k^{th} regions of the subdivision lie in $\mathcal{J}_{jk}$. This theorem actually goes further to say that any ideal-difference conditions can be translated back into smoothness conditions.

It is not new to define splines by ideal-difference conditions. For example, Proposition 1.2 in Billera–Rose [4] is a weaker version of the forward direction of Theorem 3.1. However, the result here completely identifies smoothness conditions and ideal-difference conditions.

The translation into commutative algebra is completed by Theorem 3.2. Here we show that the set of splines satisfying a family of smoothness conditions can be identified with an ideal M in the ring $\mathbb{R}[x_1, \ldots, x_n][y_1, \ldots, y_s]/\langle y_1, \ldots, y_s\rangle^2$, where s is the number of regions in the subdivision. Furthermore, this identification sends a degree d spline to a degree $d+1$ polynomial.

Sections 4 and 5 are more technical. They justify some of the steps in the software implementation of the method. For example, Lemma 4.1 describes how to obtain generators for the spline module by lifting M from the quotient to its preimage ideal $\widetilde{M}$ in the polynomial ring $\mathbb{R}[x_1, \ldots, x_n][y_1, \ldots, y_s]$. Lemmas 5.1 and 5.2 justify replacing M with a certain leading-term ideal in the Hilbert series computation.

Next, we present two applications. The first application, presented in Section 5, concerns the Hilbert series. Computing the dimensions of spline spaces with a given maximal total degree is notoriously difficult, even in dimension $n = 2$ (see e.g. [14, Chapter 9]). However, when translated into commutative algebra, it can be interpreted as determining the Hilbert series of a certain ideal. Nontrivial techniques regarding the computation of Hilbert series have been around at least since Macaulay's work in the early 1900's [16]. The SAGE implementation (described in Section 7) allows one to verify many of the spline dimension formulas conjectured in Foucart–Sorokina [11] and to extend the work of Schumaker–Wang [20] beyond continuous splines by producing dimension formulas for subdivisions with hanging vertices.

In Section 6, the second application addresses the question whether splines form a free module. This question was first asked by Billera–Rose [5], and a conjecture relating freeness and smoothness conditions was made by Rose [19]. In 2001, Dalbec–Schenck [8] suggested a counterexample to Rose's conjecture. Unfortunately, little work has been done since. It is still completely unknown when the algebra of spline functions is free over polynomials on the base. With the help of computer implementations of the Logar–Sturmfels algorithm [15] by Barwick–

Stone [3] and Fabiańska [9], the routines here provide a means to systematically investigate the question of freeness.

Finally, we have chosen to put the proofs of the mathematical statements in a section of their own. This separates them from the main text and facilitates the use of the paper as a reference for those using the computational techniques and implementations in their own research.

2 Basic Objects

Consider a domain, $\Omega \subseteq \mathbb{R}^n$, and a fixed subdivision, $\Sigma = \{\sigma_1, \ldots, \sigma_s\}$, of Ω by closed sets. A spline over Ω with respect to Σ is an assignment of an n-variable polynomial, g_j, to each region, σ_j, such that there is piecewise function $\Omega \to \mathbb{R}$ whose value on σ_j is g_j. For example, in a subdivision known as the Alfeld split [1], Ω is the domain bounded by the simplex with vertices

$$\{e_1, e_2, \ldots, e_n, -e_1 - e_2 - \cdots - e_n\} \subset \mathbb{R}^n,$$

where e_i is the i^{th} standard basis vector, and Σ is made up of regions

$$\sigma_j = \{t_1 e_1 + t_2 e_2 + \cdots + t_{n+1} e_{n+1},\ t_i \geq 0 \text{ for all } i,\ t_1 + t_2 + \cdots + t_{n+1} \leq 1,\ \text{and } t_j = 0\}$$

where we denote $-e_1 - e_2 - \cdots - e_n$ by e_{n+1}. We write $\mathfrak{A}_n$ for the n-dimensional Alfeld split.

An example of a spline function over the 2-dimensional Alfeld split is:

$$(\mathfrak{A}_2) \qquad G(x_1, x_2) = \begin{cases} x_1 & \text{for } (x_1, x_2) \in \sigma_1 \\ x_2 & \text{for } (x_1, x_2) \in \sigma_2 \\ 0 & \text{for } (x_1, x_2) \in \sigma_3 \end{cases} .$$

We denote this spline by

$$G = (x_1, x_2, 0).$$

One need not limit themselves to such prosaic examples. Any imaginable subdivision is possible. For instance, one could consider set $\Sigma = \{\sigma_{\delta_1,(a_1,b_1)}, \ldots, \sigma_{\delta_s,(a_s,b_s)}\}$ of horizontal (along x_1 axis) and vertical (along x_2 axis) dowel shapes in $\mathbb{R}^3$:

$$\begin{aligned} \sigma_{h,(a,b)} &= \{(x_1, x_2, x_3) \mid 8\sqrt{(x_1 - a)^2 + x_3^2} \leq (x_2 - b) - (x_2 - b)^2\} \text{ and} \\ \sigma_{v,(a,b)} &= \{(x_1, x_2, x_3) \mid 8\sqrt{(x_2 - b)^2 + x_3^2} \leq (x_1 - a) - (x_1 - a)^2\} \end{aligned}$$

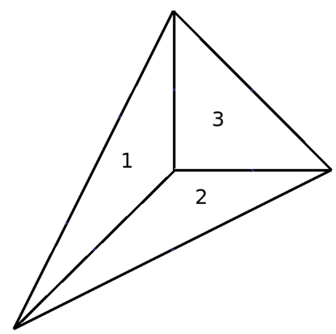

Figure 1: 2-dimensional Alfeld split with regions labelled.

and take $\Omega = \cup_i \sigma_i$ to give a configuration as in Figure 2. With this, one could entertain themselves recreating the motion of Reuben Margolin's Square Wave [17] using triples of linear splines which preserve the distance between the ends of the "dowels".

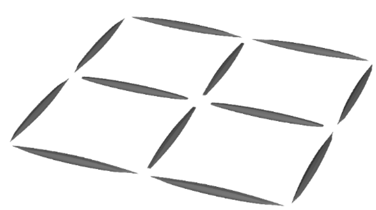

Figure 2: $\Omega =$ Margolin Square Wave subdivision

Typically, we are interested in splines which satisfy some smoothness properties. More precisely, at a point $p \in \Omega$, one can restrict attention to those splines on which any element of a given set, $\mathcal{D}_p$, of linear differential operators has a well-defined value. A choice at each point of Ω of linear differential operators is called a family of smoothness conditions if it satisfies the property:

> Given a real-valued function h on Ω, whenever Dh is well defined for all $p \in \Omega$ and all operators $D \in \mathcal{D}_p$, then so is $D(f \cdot h)$ for any $f \in \mathbb{R}[x_1, \ldots, x_n]$.

For example, one may consider splines which define C^r functions on Ω. In this case, the operators at any given point are those in the span of

$$\{\partial_{x_1}^{\alpha_1}\cdots\partial_{x_n}^{\alpha_n}|_p,\ \alpha_1+\cdots+\alpha_n\leq r\}.$$

In the case of the 2-dimensional Alfeld split above, the spline ($\mathfrak{A}_2$) is not a C^1-spline. However, the following spline is:

$$(C^1(\mathfrak{A}_2)) \qquad (x_1^4+2x_1^3x_2+x_1x_2,\ 8x_1^2x_2^2-6x_1x_2^3+x_2^4+x_1x_2,\ x_1x_2).$$

Smoothness conditions defined this way can also specify so-called "super-smoothness." For example, setting

$$\mathcal{D}_p=\begin{cases}\{\partial_{x_1}^{\alpha_1}\cdots\partial_{x_n}^{\alpha_n}|_p,\ \alpha_1+\cdots+\alpha_n\leq 1\} & \text{for } p\neq 0\\ \{\partial_{x_1}^{\alpha_1}\cdots\partial_{x_n}^{\alpha_n}|_p,\ \alpha_1+\cdots+\alpha_n\leq 3\} & \text{for } p=0\end{cases}$$

produces splines which are C^1 everywhere, and C^3 at the origin.

3 Conversion to Commutative Algebra

We denote the ring $\mathbb{R}[x_1,\ldots,x_n]$ of n-variable polynomials by R, and the set of spline functions which satisfy a family of smoothness conditions by $\mathcal{S}$. The set $\mathcal{S}$ forms an R-algebra, and by adding formal variables $y_1,\ldots,y_s$ to the ring R, $\mathcal{S}$ can be described entirely in terms of certain ideals in $R[y_1,\ldots,y_s]$. Furthermore, we can identify $\mathcal{S}$ with an ideal in the ring $R[y_1,\ldots,y_s]/\langle y_1,\ldots,y_s\rangle^2$. Ultimately this allows us to systematically approach questions about spline functions using Gröbner bases and routines that rely on them.

In the theorem below, we denote by $\mathcal{I}(\sigma_j\cap\sigma_k)=\{f\in R\mid f(p)=0\ \text{ for all } p\in\sigma_j\cap\sigma_k\}$ the set of polynomials that vanish on the intersection of σ_j and σ_k.

Theorem 3.1 *The set of splines satisfying a family of smoothness conditions forms an R-algebra and there exist ideals $\mathcal{J}_{jk}\subseteq\mathcal{I}(\sigma_j\cap\sigma_k)$ such that an s-tuple of polynomials $G=(g_1,\ldots,g_s)$ is in this ring if and only if*

$$(\text{IC}) \qquad g_j-g_k\in\mathcal{J}_{jk}\quad \textit{for all } 1\leq j,k\leq s.$$

Conversely, given any family of ideals, $\mathcal{J}_{jk}$, such that $\mathcal{J}_{jk} \subseteq \mathcal{I}(\sigma_j \cap \sigma_k)$, there is a corresponding family of smoothness conditions whose splines are determined by the equations (IC).

As an example, consider the Alfeld split. A C^r-spline $G = (g_1, \ldots, g_{n+1})$ over the Alfeld split must satisfy (IC) for the ideals

$$\mathcal{J}_{jk} = \langle \det([e_1| \cdots |\widehat{e_j}| \cdots |\widehat{e_k}| \cdots |e_{n+1}|\ x\ ])\ \rangle^{r+1}$$

where $e_{n+1} = -e_1 - e_2 - \cdots - e_n$, $x = [x_1, \ldots, x_n]^\top$, and $\widehat{\ }$ indicates omission of the column. Notice that the smoothness condition is packaged into the ideal by taking the $(r+1)^{\text{th}}$ power of the ideal $\langle \det([e_1| \cdots |\widehat{e_j}| \cdots |\widehat{e_k}| \cdots |e_{n+1}|\ x\ ])\ \rangle$. This observation relating ideal powers and smoothness was first made in Proposition 1.2 of [4].

The other result needed to study $\mathcal{S}$ as a classical commutative algebra object is the following.

Theorem 3.2 *The R-algebra $\mathcal{S}$, as an R-module, is isomorphic to the module*

$$(M) \quad M = \bigcap_{jk} (\mathcal{J}_{jk} \cdot \langle y_j + y_k \rangle + \langle y_1, \ldots, \widehat{y}_j, \ldots, \widehat{y}_k, \ldots, y_s \rangle) \quad \subseteq \quad R[y_1, \ldots, y_s]/\langle y_1, \ldots, y_s \rangle^2,$$

where $\widehat{\ }$ indicates omission of the variable.

The R-module isomorphism between $\mathcal{S}$ and M makes the identification

$$(g_1, \ldots, g_s) \leftrightarrow g_1 y_1 + \cdots + g_s y_s.$$

4 Generating Splines

The most fundamental computation that has been implemented produces a list of splines which generate $\mathcal{S}$ as an R-module. The term "generate" here is the usual sense for modules, i.e., a set of generators of $\mathcal{S}$ is a set of splines

$$\{G_1 = (g_{11}, g_{12}, \ldots, g_{1s}),\ G_2 = (g_{21}, g_{22}, \ldots, g_{2s}), \ldots,\ G_\ell = (g_{\ell 1}, g_{\ell 2}, \ldots, g_{\ell s})\} \subseteq \mathcal{S}$$

such that any spline $G \in \mathcal{S}$ can be expanded as

(SG) $$G = c_1G_1 + c_2G_2 + \cdots + c_\ell G_\ell$$

for (not necessarily unique) polynomials $c_1, \ldots, c_\ell \in R$.

For example, splines which generate the C^1-splines over the 2-dimensional Alfeld split are

($\mathcal{S}^1(\mathfrak{A}_2)$-gens) $$\{\ G_1 = (0, x_1^2x_2^2 - 2x_1x_2^3 + x_2^4, 0),\quad G_2 = (x_1^3, 3x_1x_2^2 - 2x_2^3, 0),\\ G_3 = (x_1^2x_2, 2x_1x_2^2 - x_2^3, 0),\quad G_4 = (1,1,1)\qquad \}$$

and the spline ($C^1(\mathfrak{A}_2)$) expands as

$$(x_1^4 + 2x_1^3x_2 + x_1x_2, 8x_1^2x_2^2 - 6x_1x_2^3 + x_2^4 + x_1x_2, x_1x_2) = 3G_1 + (x_1 + x_2)G_2 + x_1G_3 + x_1x_2G_4.$$

Generators for $\mathcal{S}$ appear as the coefficient vectors of the y-linear terms of a generating set for $\widetilde{M}$. Explicitly, we have the following result (Recall $M = \widetilde{M}/\langle y_1, \ldots, y_s\rangle^2$).

Lemma 4.1 *Let B denote any generating set for the ideal $\widetilde{M}$. For each element $b \in B$, let b_1 denote the y-linear term. Then the image of the set*

$$\{b_1,\ b \in B\}$$

under the map $\widetilde{M} \twoheadrightarrow M \xrightarrow{\sim} \mathcal{S}$ generates $\mathcal{S}$ as an R-module.

Theorem 3.2 and Lemma 4.1 imply that we can find a set of generating splines by computing a Gröbner basis for the intersection

($\widetilde{M}$) $$\begin{aligned}\widetilde{M} &= < y_1, y_2, \ldots, y_s >^2 + \textstyle\bigcap_{jk}(\mathcal{J}_{jk} \cdot \langle y_j + y_k\rangle + \langle y_1, \ldots, \widehat{y_j}, \ldots, \widehat{y_k}, \ldots, y_s\rangle)\\ &\subseteq R[y_1, \ldots, y_s],\end{aligned}$$

where $\widehat{\ }$ indicates omission.

5 The Hilbert Series

Another computation that has been implemented produces the Hilbert series of a given ring of splines. We denote those splines $G = (g_1, \ldots, g_s)$ satisfying $\deg g_i \leq d$ for all $1 \leq i \leq n$ by

$\mathcal{S}_d$. When $\mathcal{S}$ is the set of C^r splines over Σ, the traditional notation in spline theory (see [14, Chapter 5]) is $\mathcal{S}^r_d(\Sigma)$. For example,

$$(x_1^3, 3x_1x_2^2 - 2x_2^3, 0), (x_1^2x_2, 2x_1x_2^2 - x_2^3, 0), (1, 1, 1) \in \mathcal{S}^1_3(\mathfrak{A}_2).$$

Given any space of polynomials, we also denote the subspace made up of polynomials of degree $\leq d$ by a subscript d.

The Hilbert series is the formal power series

$$H_{\mathcal{S}}(t) = (1-t) \cdot \sum_{d \geq 0} (\dim_{\mathbb{R}} \mathcal{S}_d) t^d.$$

Its computation is particularly relevant in spline theory, because the generating function of the sequence $(\dim_{\mathbb{R}} \mathcal{S}_d)_{d\geq 0}$ of splines over a region in $\mathbb{R}^n$ takes the form (see [4])

$$(P) \qquad \sum_{d \geq 0} (\dim_{\mathbb{R}} \mathcal{S}_d) t^d = \frac{P(t)}{(1-t)^{n+1}}$$

for some polynomial $P(t) = \sum_{k=0}^{k^\star} a_k t^k$ with integer coefficients. Thus, the computation of $H_{\mathcal{S}}$ determines P and in turn yields the dimension formula

$$(\mathrm{DF}) \qquad \dim_{\mathbb{R}} \mathcal{S}_d = \sum_{k=0}^{k^\star} a_k \binom{d+n-k}{n}.$$

For example, for C^1-splines over the 2-dimensional Alfeld split, the Hilbert series and dimension formula are

$$H_{\mathcal{S}^1(\mathfrak{A}_2)}(t) = \frac{1+2t^3}{(1-t)^2}, \qquad \dim_{\mathbb{R}} \mathcal{S}^1_d(\mathfrak{A}_2) = \binom{d+2}{2} + 2\binom{d-1}{2}.$$

In the technical discussion below, it is easier to work with the module M, so we take note of the equality

$$H_M(t) = H_{\mathcal{S}}(t) \cdot t.$$

In the implementation of the computation of the Hilbert series[1], we find the series

$$H_{\widetilde{M}}(t) \text{ and } H_{\langle y_1,\ldots,y_s\rangle^2}(t)$$

[1]When asked to produce the Hilbert series of an ideal, $I \subseteq R[y_1, \ldots, y_s]$, SAGE returns the series for the dimensions of the quotient, $R[y_1, \ldots, y_s]/I$. So our implementation uses the equality: $H_I(t) = H_{R[y_1,\ldots,y_s]}(t) - H_{R[y_1,\ldots,y_s]/I}(t)$.

and use the following equality.

Lemma 5.1

$$H_M(t) = H_{\widetilde{M}}(t) - H_{\langle y_1,\dots,y_s\rangle^2}(t).$$

We use a leading-term ideal of $\widetilde{M}$ to compute $H_{\widetilde{M}}(t)$ rather than $\widetilde{M}$ itself. This change is justified by the fact that the Hilbert series depends only on the dimensions of the modules $\widetilde{M}_d$ and the following result.

Lemma 5.2 *Let $\leq$ be a monomial order on $R[y_1,\dots,y_s]$ such that, for any $f \in R[y_1,\dots,y_s]_d$, $\deg(f) = \deg(\mathrm{lt}(f))$, and let $\mathcal{L}'$ denote the leading-term ideal of an ideal $\mathcal{L}$ with respect to $\leq$, then*

$$\dim_{\mathbb{R}} \mathcal{L}_d = \dim_{\mathbb{R}} \mathcal{L}'_d.$$

The only reason for replacing $\widetilde{M}$ with a leading-term ideal is because SAGE quickly and efficiently computes Hilbert series of homogeneous ideals, but refuses to perform the computation for inhomogeneous ideals.

Verification of Foucart–Sorokina's conjectures

In [11], Foucart–Soronkina made conjectures for dimension formulas of the form (DF) of a number of tetrahedral subdivisions. The method was also based on the determination, for a fixed smoothness r, of the polynomial P in (P). The coefficients were derived one by one from the computation of $\dim_{\mathbb{R}} \mathcal{S}^r_d(\Sigma)$ for $d = 0, 1, 2, \dots$ using Alfeld's 3-dimensional applet [2]. The argument there depends on the ability to compute the dimensions up to $d = 8r + 4$. Computational limitations prevented them from accessing examples with $r > 3$.

The method presented here works similarly, however moving to the leading-term ideal in Lemma 5.2 vastly increases the speed at which these dimensions can be computed. For instance, the Hilbert series for the 3-dimensional Alfeld split with $r = 3$ and $r = 4$, namely

$$H_{\mathcal{S}^3(\mathfrak{A}_3)} = \frac{1+3t^8}{(1-t)^3}, \qquad H_{\mathcal{S}^4(\mathfrak{A}_3)} = \frac{1+t^9+t^{10}+t^{11}}{(1-t)^3},$$

were computed in less than a second while the method of [11] required several days. No careful performance assessment was made, but in this example there was at least a thousand-fold increase in speed.

These computations support the conjectured formula in [11] for not only the 3-dimensional Alfeld split, but for the arbitrary dimensional case. We have also verified the formulas they suggested for other subdivisons, and invite the reader to do the same with our implementations [7, 10].

Non-simplicial subdivisions

As we have pointed out, there is no limit to the kinds of subdivisions we can consider. For instance, we can consider "triangulations" with hanging vertices, as in Figure 3. This is not a simplicial complex since neither the intersection of the 1^{st} and 2^{nd} nor the intersection of the 1^{st} and 3^{rd} regions are facets of the 1^{st} region. The dimension of the space of continuous splines

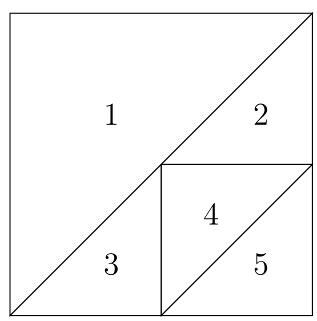

Figure 3: Example of a triangular subdivision with one hanging vertex.

over such triangulations is given in [20, Theorem 5.3] by $\dim \mathcal{S}_0^0(\Sigma) = 1$ and for $d \geq 1$ by

$$\dim \mathcal{S}_d^0(\Sigma) = V_{NH} + E_c(d-1) + N_T \binom{d-1}{2}.$$

Here, V_{NH}, E_c and N_T are the numbers of non-hanging vertices, composite edges, and triangles, respectively. In particular, for the example in Figure 3, the dimension formula is

$$\dim \mathcal{S}_d^0(\Sigma) = 6 + 10(d-1) + 5\binom{d-1}{2}, \qquad d \geq 1.$$

Our implementation gives the Hilbert series and the dimension formula (in a different form than above)

$$H_{\mathcal{S}^0(\Sigma)}(t) = \frac{1+3t+t^2}{(1-t)^2}, \qquad \dim_{\mathbb{R}} \mathcal{S}^0_d(\Sigma) = \binom{d+2}{2} + 3\binom{d+1}{2} + \binom{d}{2}.$$

More generally, computations for $0 \le r \le 12$ and extrapolation of the behavior in r suggest that, for any $r \ge 0$,

$$H_{\mathcal{S}^r(\Sigma)}(t) = \frac{1+2t^{r+1}+t^{\frac{3r}{2}+1}+t^{\frac{3r}{2}+2}}{(1-t)^2}, \quad r \text{ even}, \qquad H_{\mathcal{S}^r(\Sigma)}(t) = \frac{1+2t^{r+1}+2t^{\frac{3(r+1)}{2}}}{(1-t)^2}, \quad r \text{ odd}.$$

6 The Question of Freeness

Another interesting implementation is one which determines if $\mathcal{S}$ is free over R, and if so computes a "free basis". In some situations, it may be sufficient to simply know whether or not $\mathcal{S}$ admits a free basis. In other situations, it may be desirable to replace a given generating set $\{G_1, \cdots, G_m\}$ with the free basis $\{G'_1, \dots, G'_\ell\}$.

Recall that the ring of splines, considered as an R-module, is called free if there are generators $\{G'_1, \dots, G'_\ell\}$ for which the expansion in (SG) is unique for any $G \in \mathcal{S}$:

$$G = c_1 G'_1 + c_2 G'_2 + \cdots + c_\ell G'_\ell.$$

In this case, $\{G'_1, \dots, G'_\ell\}$ is called a free basis.

For example, the collection ($\mathcal{S}^1(\mathfrak{A}_2)$-gens) generates the C^1-splines over the 2-dimensional Alfeld split. However, these generators do not give unique expansions. For instance, there is the following non-trivial relation:

$$2G_1 + x_2 G_2 - x_1 G_3 + 0 G_4 = 0.$$

On the other hand, a free basis is given by the generating set

$$\{G'_1 = (x_1^3, 3x_1x_2^2 - 2x_2^3, 0), G'_2 = (x_1^2 x_2, 2x_1x_2^2 - x_2^3 0), G'_3 = (1, 1, 1)\}.$$

Not all spline rings admit a free basis. An interesting example is that of Billera–Rose [5] in which they consider C^2-splines over a deformed octahedron with vertices

$$(1,0,0),\ (0,1,0),\ (1,1,1),\ (-1,0,0),\ (0,-1,0),\ (0,0,-1)$$

and subdivided into the eight tetrahedra formed as a convex combination of a facet and the origin.

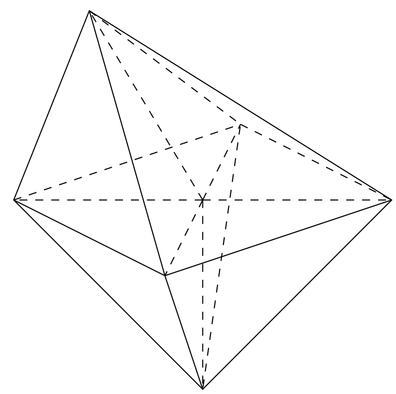

Figure 4: Billera–Rose non-free example.

The mathematics to determine if $\mathcal{S}$ has a free basis comes from the Quillen–Suslin theorem [18, 22]. This theorem guarantees that $\mathcal{S}$ is free if and only if it is "locally free." Once it is known that $\mathcal{S}$ is free, the algorithm of Logar–Sturmfels [15] allows a free basis to be computed from an arbitrary generating set.

Practically speaking, this means the following: Assume generators $\{G_1, \ldots, G_m\}$ have been obtained via the approach used in Section 4. Denote by $\mathcal{G}$ the $s \times m$ matrix whose n^{th} column contains the entries of G_n, and denote by $\mathcal{R}$ the matrix of column relations of $\mathcal{G}$, i.e.

$$\mathcal{G} \cdot \mathcal{R} = 0.$$

Let μ be the largest number for which there is a non-zero μ-minor of $\mathcal{G}$ (i.e., a $\mu \times \mu$ submatrix with non-zero determinant). Then $\mathcal{S}$ is locally-free if and only if we have the equality of ideals

$$\langle 1 \rangle = \langle f \mid f \text{ is a } \mu\text{-minor of } \mathcal{R} \rangle.$$

This equality can be checked using a Gröbner basis calculation.

Once it is known that $\mathcal{S}$ is locally free, a generating set for which expansions are unique exists. Versions of the Logar–Sturmfels algorithm to find a free basis have been implemented by Barwick–Stone [3] and Fabiańska [9].

Rose's Conjecture

In [8], Dalbec–Schenck suggested a counter-example to a conjecture of Rose [19]. We have made the surprising discovery that the splines in the Dalbec–Schenck example do not contradict the Rose's conjecture.

To be precise, Rose's conjecture reads as follows.

Conjecture 6.1 *Consider a triangulation, Σ, of a topological n-ball, Ω, in $\mathbb{R}^n$. There is a positive integer r_0 such that the C^r-splines are free if and only if $r < r_0$.*

Dalbec–Schenck suggested that the ring of C^r-splines over

$$\Sigma = \{\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6\} \subseteq \mathbb{R}^3$$

where each σ_j is a convex hull of four vertices, namely

$$\begin{aligned}
\sigma_1 = &\quad \mathrm{conv}\{(0,0,0),(3,0,0),(0,3,0),(0,0,3)\}\\
\sigma_2 = &\quad \mathrm{conv}\{(0,0,0),(3,0,0),(0,3,0),(1,0,-3)\}\\
\sigma_3 = &\quad \mathrm{conv}\{(0,0,0),(-2,-2,1),(0,0,3),(3,0,0)\}\\
\sigma_4 = &\quad \mathrm{conv}\{(0,0,0),(-2,-2,1),(0,0,3),(0,3,0)\}\\
\sigma_5 = &\quad \mathrm{conv}\{(0,0,0),(-2,-2,1),(1,0,-3),(3,0,0)\}\\
\sigma_6 = &\quad \mathrm{conv}\{(0,0,0),(-2,-2,1),(1,0,-3),(0,3,0)\}
\end{aligned}$$

is free for $r \leq 4$ and $r = 6$, yet not free for $r = 5$.

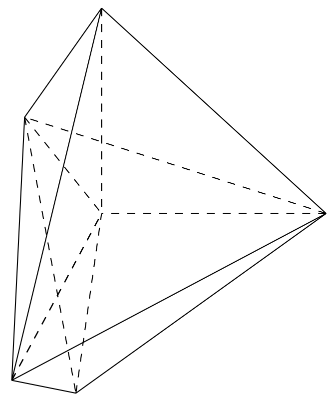

Figure 5: Dalbec–Schenck example.

7 Implementation

This section describes several routines implemented in the files `spline_routines.sage` [7] and `SplineDim` [10] using the algorithms described above. `spline_routines.sage` is a collection of small SAGE routines which serve as a "proof-of-concept" of the computational techniques. The first subsection focuses on few of the most useful routines in `spline_routines.sage` and provides an example of utilization. Passing appropriate input into the routine `poly_splines(poly_list, r)` requires some moderate care. So we dedicate some additional explanation of its usage.

The second subsection presents a collection of routines [10] designed specifically for the computation of dimensions in spline theory. These routines are "pre-packaged" so that one can easily jump in and begin using the software without time spent on trying to figure out how to use `spline_routines.sage`.

7.1 `spline_routines.sage`

`poly_splines(poly_list, r)`:
This routine returns the commutative algebraic data necessary to compute C^r splines of a configuration of polyhedral regions. This data is in the form of a tuple `(R, s, J)` where `R` is

a polynomial ring from which the spline entries come, `s` is the number of polyhedral regions, and `J` is a function which takes $0 \leq j, k < s$ and returns the ideal of R giving the condition in equation (IC).

The following two routines take input in the form `(R, s, J)`. For instance, one can produce this data from `poly_splines(poly_list, r)`, or simply code it manually[2].

`spline_Hilbert_series(R, s, J)`:
This routine returns the Hilbert series as a rational function.

`spline_module_generators(R, s, J)`:
This routine returns a list of spline module generators.

The last two routines take a list of spline module generators. This is usually obtained from running `spline_module_generators(R, s, J)`.

`is_projective(smg)`:
This returns true if the spline ring is free, and false otherwise.

`compute_free_basis(smg)`:
If the spline module is free, this routine returns a free basis of splines module generators.

As an example, we show the computation of the C^1-splines over the T-mesh in Figure 6.[3]

This code should be run in SAGE, from a directory containing `spline_routines.sage`. SAGE should be able to access Macaulay2 [12], and Barwick–Stone package, `QuillenSuslin.m2`, should be on Macaulay2's load-path.

[2]The possible subdivisions and smoothness conditions these routines can accept as input are limited only by what the user is willing to manually code.

[3]Splines over T-meshes as discussed here were considered in [13] — we stress that these are not the so-called T-splines

```
load spline_routines.sage

R1 = ((0,0),(1,0),(1,1),(1,2),(0,2))
R2 = ((1,1),(2,1),(2,2),(1,2))
R3 = ((2,1),(3,1),(3,2),(2,2))
R4 = ((1,0),(3,0),(3,1),(2,1),(1,1))


poly_list = (R1, R2, R3, R4)

(R, s, J) = poly_splines(poly_list , 1)

sHs = spline_Hilbert_series(R, s, J); print sHs

smg = spline_module_generators(R, s, J);
sm_is_free = is_projective(smg)

if sm_is_free:
    print 'splines are free'
    free_smg = compute_free_basis(smg)
    print free_smg
else:
    print 'splines are not free'
    print smg
```

This gives the output:

```
(2*t^4 + t^2 + 1)/(t^2 - 2*t + 1)
splines are free
[
(0, 0, y0^2*y1^2 - 2*y0^2*y1 - 4*y0*y1^2 + y0^2 + 8*y0*y1 +
                                     4*y1^2 - 4*y0 - 8*y1 + 4, 0),
(0, 0, 0, y0^2*y1^2 - 2*y0^2*y1 - 2*y0*y1^2 + y0^2 + 4*y0*y1 +
                                         y1^2 - 2*y0 - 2*y1 + 1),
(0, y0^2 - 2*y0 + 1, y0^2 - 2*y0 + 1, y0^2 - 2*y0 + 1),
(1, 1, 1, 1)
]
```

To investigate other examples, one need only modify `poly_list` above.

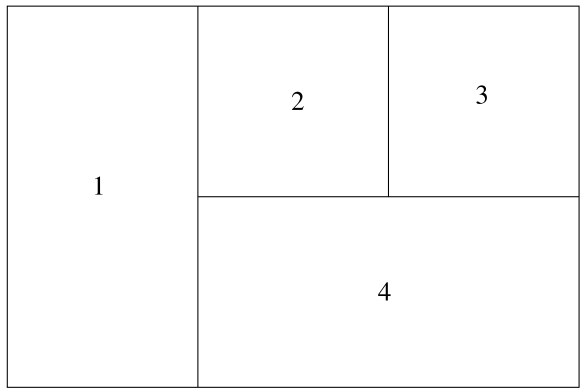

Figure 6: T-mesh.

We have also created a file, `examples.sage` [6], which contains many examples. This includes all the examples here and more, as well as a routine to check Rose's conjecture on the Dalbec–Schenck example.

The proper use of `poly_splines(poly_list, r)`**:**

Each polyhedral region in `poly_list` is specified by points on its perimeter. This should include any "hanging vertices," as in Figure 3. It is not required that the specified points determine the polyhedral region. However, this routine assumes that intersection of any two polyhedra in the list equals the closure of an open set in the affine linear space spanned by the specified points that the polyhedra have in common.

Configurations made up of convex polyhedral sets can always be described in this way. For example,

```
poly_list = (
((1,0),(1,1),(1,2)),
((1,1),(2,1),(2,2),(1,2)),
((2,1),(3,1),(2,2)),
((1,0),(3,1),(2,1),(1,1)) )
```

is another acceptable description of the T-mesh in Figure 6.

Further, it is possible to describe some configurations in which the polyhedra are not necessarily convex, for example, the configuration in Figure 7 (left). However, splines over the configuration in Figure 7 (right) cannot be described using this routine.

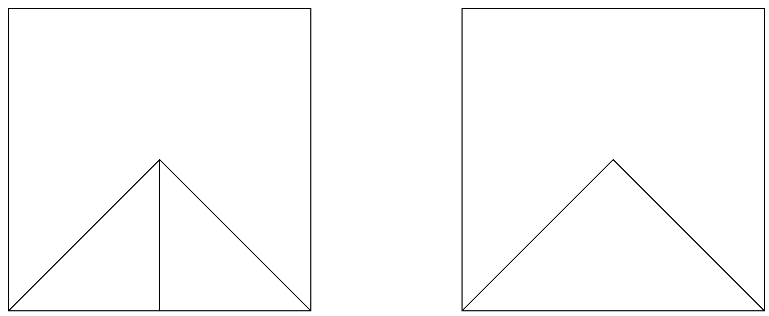

Figure 7: Left: configuration acceptable directly into `poly_splines(poly_list, r)`; Right: configuration requiring manual input of the `J` function.

7.2 `SplineDim`

Building on the previous implementation, we have created a tool designed specifically for spline theorists investigating dimension formulas. The resulting software, called `SplineDim`, consists of a collection of routines written in SAGE and of files containing a number of predefined subdivisions. It is available online at [10], where one also finds a user's guide complementing the following short descriptions of the main routines.

`spline_gf(Sigma, r)`:
This routine returns the generating function of the sequence $\left(\dim_{\mathbb{R}} \mathcal{S}_d^r(\Sigma)\right)_{d\geq 0}$ in the form (P).

`gf_to_dims(GF, d)`:
Given a generating function in the form (P), this routines outputs a list of $d+1$ values for the dimensions corresponding to degrees $0, 1, \ldots, d$.

`spline_dims(Sigma, r, d)`:
This routine combines the two previous ones without making a reference to the generating

function when returning the values for $\dim_{\mathbb{R}} \mathcal{S}^r_0(\Sigma), \dim_{\mathbb{R}} \mathcal{S}^r_1(\Sigma), \ldots, \dim_{\mathbb{R}} \mathcal{S}^r_d(\Sigma)$. If only the value for $\dim_{\mathbb{R}} \mathcal{S}^r_d(\Sigma)$ is sought, one can use `spline_dim(Sigma, r, d)`.

8 Proofs

Theorem 3.1 *The set of splines satisfying a family of smoothness conditions forms an R-algebra, and there exist ideals $\mathcal{J}_{jk} \subseteq \mathcal{I}(\sigma_j \cap \sigma_k)$ such that an s-tuple of polynomials $G = (g_1, \ldots, g_s)$ is in this ring if and only if*

$$\text{(IC)} \qquad g_j - g_k \in \mathcal{J}_{jk} \quad \text{for all } 1 \leq j, k \leq s.$$

Conversely, given any family of ideals, $\mathcal{J}_{jk}$, such that $\mathcal{J}_{jk} \subseteq \mathcal{I}(\sigma_j \cap \sigma_k)$, there is a corresponding family of smoothness conditions whose splines are determined by the equations (IC).

Proof Begin with a family of smoothness conditions $\mathcal{D}$ on Ω. Consider DG for a differential operator $D \in \mathcal{D}_p$ and a spline function $G = (g_1, \ldots, g_s) \in R \times \cdots \times R$. For DG to be well defined at p, we must have

$$\text{(WD)} \qquad Dg_j = Dg_k$$

whenever $p \in \sigma_j \cap \sigma_k$. Equivalently,

$$D(g_j - g_k) = 0.$$

The definition of a smoothness condition requires that

$$D(f(g_j - g_k)) = 0$$

for all $f \in R$. Since D is linear this means $\widetilde{\mathcal{J}}_D = \{h \in R \mid Dh = 0\}$ is an ideal, and our equation (WD) is equivalent to requiring that $g_j - g_k$ must lie in $\widetilde{\mathcal{J}}_D$ whenever $p \in \sigma_j \cap \sigma_k$. Furthermore, G must define function on Ω, so $g_j - g_k \in \mathcal{I}(\sigma_j \cap \sigma_k)$.

Set $\mathcal{J}_D = \widetilde{\mathcal{J}}_D \cap \mathcal{I}(\sigma_j \cap \sigma_k)$, and denote

$$\mathcal{J}_{jk} = \bigcap_{p \in \sigma_j \cap \sigma_k} \bigcap_{D \in \mathcal{D}_p} \mathcal{J}_D,$$

then we end up with the requirement

$$g_j - g_k \in \mathcal{J}_{jk}.$$

Conversely, begin with a family of ideals $\mathcal{J}_{jk} \subseteq \mathcal{I}(\sigma_j \cap \sigma_k)$. Denote by $\mathcal{S}$ the subring of $R \times \cdots \times R$ defined by (IC).

Observe that $\mathcal{S}$ is made up of splines because if $G = (g_1, \ldots, g_s) \in \mathcal{S}$, then $g_j - g_k \in \mathcal{I}(\sigma_j \cap \sigma_k)$. In other words, $g_j(p) = g_k(p)$ for all $p \in \sigma_j \cap \sigma_k$.

Consider $p \in \Omega$, and set

$$\text{(Jp)} \qquad \mathcal{J}_p = \bigcap_{p \in \mathbb{V}(\mathcal{J}_{jk})} \mathcal{J}_{jk},$$

where $\mathbb{V}(\mathcal{J}_{jk})$ is the set of points at which all elements of $\mathcal{J}_{jk}$ vanish. Denote by $\mathcal{D}_p$ the linear differential operators at p which annihilate $\mathcal{J}_p$. Note that $\mathcal{D}_p$ carves-out exactly $\mathcal{J}_p$:

$$\text{(DJ)} \qquad \mathcal{J}_p = \{h \in R \mid Dh = 0 \ \text{ for all } D \in \mathcal{D}_p\}.$$

With these definitions, we know that $\mathcal{S}$ is contained in the splines satisfying the smoothness conditions $\mathcal{D}$.

Now consider an arbitrary spline $G = (g_1, \ldots, g_s)$. If G is not in $\mathcal{S}$, then there is some jk such that $g_j - g_k \notin \mathcal{J}_{jk}$. This means at any $p \in \sigma_j \cap \sigma_k$, there is some $D \in \mathcal{D}_p$ such that

$$D(g_j - g_k) \neq 0,$$

i.e., DG does not have a well-defined value. ∎

Theorem 3.2 *The R-algebra $\mathcal{S}$, as an R-module, is isomorphic to the module*

$$(M) \quad M = \bigcap_{jk} (\mathcal{J}_{jk} \cdot \langle y_j + y_k \rangle + \langle y_1, \ldots, \widehat{y_j}, \ldots, \widehat{y_k}, \ldots, y_s \rangle) \quad \subseteq \quad R[y_1, \ldots, y_s]/\langle y_1, \ldots, y_s \rangle^2,$$

where $\widehat{\ }$ indicates omission of the variable.

Proof Consider the surjective map of R-modules

$$\langle y_1, \ldots, y_s \rangle / \langle y_1, \ldots, y_s \rangle^2 \twoheadrightarrow \underbrace{R \times \cdots \times R}_{s-\text{copies}}$$

which sends $g_1 y_1 + \cdots + g_s y_s$ to $(g_1, \ldots, g_s)$. The condition $g_j - g_k \in \mathcal{J}_{jk}$ is exactly satified by elements of

$$\mathcal{J}_{jk} \cdot \langle y_j + y_k \rangle + \langle y_1, \ldots, \widehat{y}_j, \ldots, \widehat{y}_k, \ldots, y_s \rangle \quad \subseteq \quad R[y_1, \ldots, y_s] / \langle y_1, \ldots, y_s \rangle^2,$$

where $\widehat{\ }$ indicates omission. ∎

Lemma 4.1 *Let B denote any generating set for the ideal $\widetilde{M}$. For each element $b \in B$, let b_1 denote the y-linear term. Then the image of the set*

$$\{b_1, \ b \in B\}$$

under the map $\widetilde{M} \twoheadrightarrow M \xrightarrow{\sim} \mathcal{S}$ generates $\mathcal{S}$ as an R-module.

Proof The image of the generator b in M equals the image of b_1. We can now apply Theorem 3.2. ∎

Lemma 5.1

$$H_M(t) = H_{\widetilde{M}}(t) - H_{\langle y_1, \ldots, y_s \rangle^2}(t).$$

Proof Dimension is additive on exact sequences. ∎

Lemma 5.2 *Let $\leq$ be a monomial order on $R[y_1, \ldots, y_s]$ such that, for any $f \in R[y_1, \ldots, y_s]_d$, $\deg(f) = \deg(\operatorname{lt}(f))$, and let $\mathcal{L}'$ denote the leading-term ideal of an ideal $\mathcal{L}$ with respect to $\leq$, then*

$$\dim_{\mathbb{R}} \mathcal{L}_d = \dim_{\mathbb{R}} \mathcal{L}'_d.$$

Proof We proceed by induction. The base case $d = 0$ is evident since either both $\mathcal{L}_0$ and $\mathcal{L}'_0$ are $\{0\}$ or both are $\mathbb{R}$.

Now consider the isomorphisms obtained from the 2nd isomorphism theorem, and the equality $\mathcal{L}_d + R_{d-1} = \mathcal{L}'_d + R_{d-1}$:

$$\begin{array}{rcl} \mathcal{L}_d/\mathcal{L}_{d-1} & \cong & (\mathcal{L}_d + R_{d-1})/R_{d-1} \\ & = & (\mathcal{L}'_{d-1} + R_{d-1})/R_{d-1} \\ & \cong & \mathcal{L}'_d/\mathcal{L}'_{d-1}. \end{array}$$

∎

References

[1] Alfeld P (1984) A trivariate Clough–Tocher scheme for tetrahedral data. Comput Aided Geom Design (1):169–181

[2] Alfeld P (2001) MDS applet and 3D MDS applet. URL http://www.math.utah.edu/~pa/

[3] Barwick B, Stone B (2012) QuillenSuslin.m2. Available at http://faculty.uscupstate.edu/bbarwick/math.html

[4] Billera LJ, Rose LL (1991) A dimension series for multivariate splines. Discrete Comput Geom 6(2):107–128, DOI 10.1007/BF02574678, URL http://dx.doi.org/10.1007/BF02574678

[5] Billera LJ, Rose LL (1992) Modules of piecewise polynomials and their freeness. Math Z 209(4):485–497, DOI 10.1007/BF02570848, URL http://dx.doi.org/10.1007/BF02570848

[6] Clarke P (2013) examples.sage. Available at http://www.math.drexel.edu/∼pclarke

[7] Clarke P (2013) spline_routines.sage. Available at http://www.math.drexel.edu/∼pclarke

[8] Dalbec JP, Schenck H (2001) On a conjecture of Rose. J Pure Appl Algebra 165(2):151–154, DOI 10.1016/S0022-4049(00)00175-4, URL http://dx.doi.org/10.1016/S0022-4049(00)00175-4

[9] Fabiańska A (2001) QuillenSuslin, version 1. Available at http://wwwb.math.rwth-aachen.de/QuillenSuslin/

[10] Foucart S (2013) SplineDim. URL http://www.math.drexel.edu/~foucart/software.htm

[11] Foucart S, Sorokina T (2013) Generating dimension formulas for multivariate splines. Albanian J Math 7(1):24–35, URL http://x.kerkoje.com/index.php/ajm/article/view/527/484

[12] Grayson DR, Stillman ME (2012) Macaulay2, a software system for research in algebraic geometry, version 1.5. Available at http://www.math.uiuc.edu/Macaulay2/

[13] Ibrahim AK (1989) Dimension of superspline spaces defined over rectilinear partitions. Approximation Theory VI pp 337–340

[14] Lai MJ, Schumaker LL (2007) Spline functions on triangulations, vol 110. Cambridge University Press

[15] Logar A, Sturmfels B (1992) Algorithms for the Quillen-Suslin theorem. J Algebra 145(1):231–239, DOI 10.1016/0021-8693(92)90189-S, URL http://dx.doi.org/10.1016/0021-8693(92)90189-S

[16] MacAulay FS (1916) Some Properties of Enumeration in the Theory of Modular Systems. Proc London Math Soc S2-26

[17] Margolin R (2005) Square Wave: Math and the Night Sea

[18] Quillen D (1976) Projective modules over polynomial rings. Invent Math 36:167–171

[19] Rose LL (2001) Unknown source

[20] Schumaker LL, Wang L (2011) Spline spaces on TR-meshes with hanging vertices. Numerische Mathematik 118(3):531–548, DOI 10.1007/s00211-010-0353-0, URL http://dx.doi.org/10.1007/s00211-010-0353-0

[21] Stein W, et al (2013) Sage Mathematics Software, version 5.8. The Sage Development Team

[22] Suslin AA (1976) Projective modules over polynomial rings are free. Dokl Akad Nauk SSSR 229(5):1063–1066