SplineDim: User's Guide*

Patrick Clarke and Simon Foucart — Drexel University

SplineDim is a collection of routines written in SAGE [3] and designed to compute dimension formulas for multivariate splines. It was written by P. Clarke and S. Foucart (mostly by P. Clarke) and is available on S. Foucart's web page at `www.math.drexel.edu/~foucart`[1].

1 A point on the underlying theory

Given an integer $n \geq 2$, given a simplicial partition Δ of a polygonal region $\Omega \in \mathbb{R}^n$, given a degree $d \geq 0$, and given a smoothness parameter $r \geq 0$, one considers the space

$$\mathcal{S}_d^r(\Delta) = \big\{ s \in \mathcal{C}^r(\Omega) : s_{|T} \text{ is a } n\text{-variate polynomial of degree} \leq d \text{ for each simplex } T \in \Delta \big\}$$

of $\mathcal{C}^r$ splines of degree $\leq d$ in n variables over Δ. One is interested in the dimension

$$\dim \mathcal{S}_d^r(\Delta).$$

P. Alfeld's applets [1, 2] have already provided tools for the computations of these quantities for fixed Δ, d, and r when $n = 2$ or $n = 3$. The present software computes the whole sequence

$$\big(\dim \mathcal{S}_d^r(\Delta) \big)_{d \geq 0}$$

for arbitrary n's (although the computational time becomes prohibitive when n becomes large). This is based (see [4] for details) on the computation of the generating function of the sequence which is known to be of the form

$$\sum_{d \geq 0} \dim \mathcal{S}_d^r(\Delta)\, t^d = \frac{P(t)}{(1-t)^{n+1}} \tag{1}$$

for some polynomial P with integer coefficients. The determination of this polynomial yields a dimension formula for fixed Δ and r, namely

$$\dim \mathcal{S}_d^r(\Delta) = \sum_{k=0}^{k^\star} a_k \binom{d+n-k}{n} \qquad \text{where} \quad P(t) = \sum_{k=0}^{k^\star} a_k\, t^k.$$

Two further quantities of interest can be derived from P, namely

$$\begin{aligned} d_\star = &\ \text{largest value such that all splines } s \in \mathcal{S}_{d_\star}^r(\Delta) \text{ are in fact polynomials,} \\ = &\ \text{largest value such that } a_k = 0 \text{ for all } k \in \{1, \ldots, d_\star\}, \\ d^\star = &\ \text{smallest value such that } \dim \mathcal{S}_d^r(\Delta) \text{ agrees with a polynomial expression in } d \text{ for } d \geq d^\star, \\ = &\ \deg(P) - n. \end{aligned}$$

*Last update: 24 April 2013

[1] Soon to be `www.math.uga.edu/~foucart`

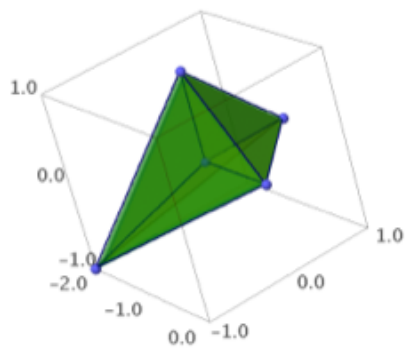

Figure 1: Alfeld split in three dimensions

2 Some Examples

It is now time to illustrate the usage of the software on a few examples, for instance by verifying some of the conjectures made in [5]. We start with the **Alfeld split in dimension three**, which is the tetrahedral partition shown on Figure 1.

First, you need to have installed SAGE on your computer. Open it and launch a terminal session. Then change the directory to the one containing SplineDimCode and load the file containing the main routines by typing

```
load spline_dim.sage
```

We start by entering the partition as vertices given by their coordinates and as simplicial regions given by their vertices. For instance, the two-dimensional Alfeld split is entered as

```
def alfeld2d():
    (v0, v1, v2, v3) = ( (0,0), (1,0), (0,1), (-1,-1) )
    (t0, t1, t2) = ( (v0, v1, v2), (v0, v2, v3), (v0, v3, v1 ) )
    return (t0, t1, t2)
```

To visualize this triangulation, type

```
D = type_triangulation(alfeld2d())
D.plot()
```

The result[2] is displayed in Figure 2.

[2] note the slight issue with the origin of coordinates

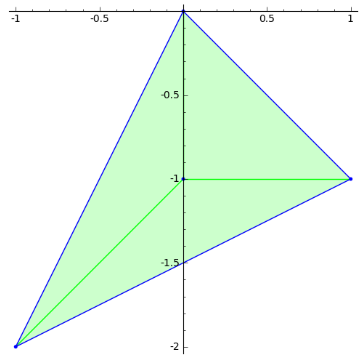

Figure 2: Alfeld split in two dimensions (aka Clough–Tocher split)

Many partitions have already been predefined: the file *triangulations.sage* contains two-dimensional partitions[3], the file *tetrahedrations.sage* contains three-dimensional partitions[4], and the file *partitions.sage* contains routines to construct general n-dimensional partitions[5]. Thus, Figure 1 could be created by typing (do not forget to `load tetrahedrations.sage`)

```
D = type_triangulation(alfeld3d())
D.plot()
```

Let us now compute dimensions for the partition Δ chosen to be the three-dimensional Alfeld split and for r fixed to be 5. These two objects are defined by typing

```
Delta = alfeld3d()
r=5
```

For a given degree, say $d = 13$, the dimension of $\mathcal{S}_d^r(\Delta)$ is obtained as

```
d=13; spline_dim(Delta,r,d)
```

The answer should be `572`. In fact, hidden behind this computation is the computation of the generating function (1), which is obtained by calling `spline_gf`. Thus, naming the output `GF_A3`, say, we type

```
GF_A3=spline_gf(Delta,r); GF_A3
```

and the output should be

```
(-3*t^12 - 1)/(-t^4 + 4*t^3 - 6*t^2 + 4*t - 1)
```

From there, the dimension calculated above can be retrieved via `gf_to_dim(GF_A3,d)`. This is recommended over `spline_dim(Delta,r,d)` when the generating function has already

[3]only a few of them at the moment

[4]with roughly the same nomenclature as in Alfeld's trivariate applet [2]

[5]only a routine for the Alfeld split has been written so far

been computed, otherwise an expensive computation is performed again. With the generating function at hand, the whole sequence of dimensions up to a given d can also be produced (d stills equals 13 in this example) directly as

```
gf_to_dims(GF_A3,d)
```

and the result should be

```
[1, 4, 10, 20, 35, 56, 84, 120, 165, 220, 286, 364, 458, 572]
```

In case the generating function is not available, the command `spline_dims(Delta,r,d)` can be used. In the above list, the dimensions corresponding to $d = 0, 1, \ldots, 11$ equal the dimensions of the space of n-variate polynomials of degree $\leq d$ (namely, $\binom{d+n}{n}$), so that $d_\star = 11$. This value can be obtained by calling

```
gf_to_dsub(GF_A3)
```

(or `dsub(Delta,r)` in case the generating function is not available). Likewise, the value of $d^\star$ can be obtained by calling `gf_to_dsup(GF_A3)` or `dsup(Delta,r)` — it should be $d^\star = 9$ here.

We now look at the **Alfeld splits in dimensions four and five**. For $n = 4$ and $r = 4$, the generating function is obtained as follows:

```
load partitions.sage
n=4; r=4;
Delta=alfeld(n);
GF_A4=spline_gf(Delta,r); GF_A4
```

The output should be

```
(t^14 + t^13 + t^12 + t^11 + 1)/(-t^5 + 5*t^4 - 10*t^3 + 10*t^2 - 5*t + 1)
```

For $n = 5$ and $r = 3$, the generating function is obtained (after a longer computational time) as follows:

```
n=5; r=3;
Delta=alfeld(n);
GF_A5=spline_gf(Delta,r); GF_A5
```

The output should be

```
(-5*t^12 - 1)/(-t^6 + 6*t^5 - 15*t^4 + 20*t^3 - 15*t^2 + 6*t - 1)
```

All these examples are consistent (as other examples would be) with the formula conjectured in [5] for the generating function relative to the Alfeld split A_n in n dimensions, namely

$$\sum_{d \geq 0} \dim \mathcal{S}_d^r(\Delta)\, t^d = \frac{P(t)}{(1-t)^{n+1}},$$

$$\text{where } P(t) = \begin{cases} 1 + n\, t^{\frac{r+1}{2}(n+1)}, & \text{if } r \text{ is odd,} \\ 1 + \sum_{j=1}^{n} t^{\frac{r}{2}(n+1)+j} & \text{if } r \text{ is even.} \end{cases}$$

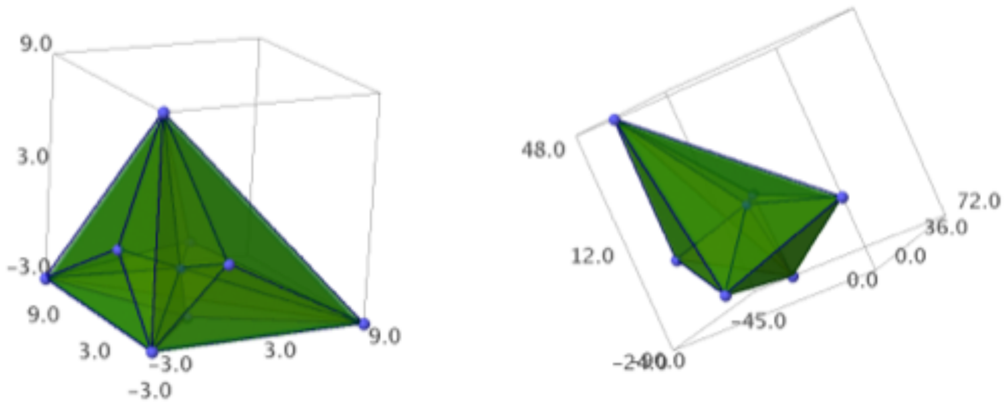

Figure 3: Worsey–Farin split (left) and generic 8-cell (right)

Finally, we examine two other partitions considered in [5] and shown on Figure 3: the **Worsey–Farin split** and the **generic** 8**-cell**. For each of them, all the dimensions up to $d = 15$ can be computed when $r = 3$ via the following command lines: for Worsey–Farin split,

```
d=15; r=3;
Delta = Worsey_Farin();
GF_WF=spline_gf(Delta,r);
gf_to_dims(GF_WF,d)
```

giving the output

```
[1, 4, 10, 20, 35, 56, 92, 152, 248, 392, 596, 872, 1232, 1688, 2252, 2936]
```

and for the generic 8-cell, without computing the generating function first,

```
Delta = generic_8cell()
spline_dims(Delta,r,d)
```

giving the output

```
[1, 4, 10, 20, 35, 56, 84, 126, 192, 290, 428, 614, 856, 1162, 1540, 1998]
```

3 Non-simplicial partitions

`SplineDim` can handle very general partitions[6], such as the triangulation with one hanging vertex depicted in Figure 4. It is advisable to enter each region by listing all the vertices on its perimeter. Thus, for the example of Figure 4,

```
Delta=(    ( (0,0), (1,1), (2,2), (0,2)),
           ( (1,1), (2,1), (2,2) ),
           ( (0,0), (1,0), (1,1) ),
           ( (1,0), (2,1), (1,1) ),
           ( (1,0), (2,0), (2,1) )        )
```

[6] a few examples have been included in *meshes.sage*

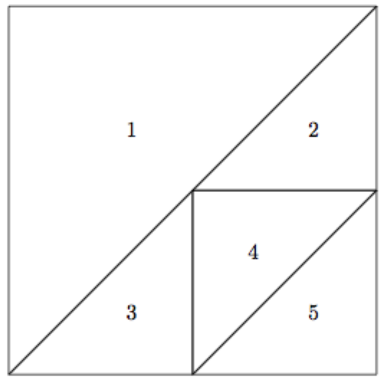

Figure 4: Example of a triangular subdivision with one hanging vertex

The dimensions and generating functions for this partition are calculated by calling the usual routines, for instance

```
r=0; d=15; spline_dims(Delta,r,d)
```

produces the output

```
[1, 6, 16, 31, 51, 76, 106, 141, 181, 226, 276, 331, 391, 456, 526, 601]
```

and

```
r=3; spline_gf(Delta,r)
```

produces the output

```
(2*t^6 + 2*t^4 + 1)/(-t^3 + 3*t^2 - 3*t + 1)
```

References

[1] P. Alfeld, MDS, www.math.utah.edu/~pa/MDS/

[2] P. Alfeld, 3D MDS, www.math.utah.edu/~pa/3DMDS/

[3] William A. Stein et al., Sage Mathematics Software (Version 5.0-OSX-64bit-10.6), The Sage Development Team, 2012, www.sagemath.org

[4] P. Clarke and S. Foucart. Symbolic spline computations. In preparation.

[5] S. Foucart and T. Sorokina. Generating dimension formulas for multivariate splines. Accepted, Albanian Journal of Mathematics.