Finding Minimal Bases in Arbitrary Spline Spaces

Ana Paula Resende Malheiro* Jorge Stolfi*

Abstract

In this work we describe a general algorithm to find a finite-element basis with minimum total support for an arbitrary spline space, given any basis for that same space. The running time is exponential on n in the worst case, but $O(nm^3)$ for many cases of practical interest, where n is the number of mesh cells and m is the dimension of the spline space.

1 Introduction

In general terms, a *spline* is a piecewise-defined function with pieces of a certain type, joined with certain smoothness constraints. Among all spline families, the *polynomial* ones are the most popular.

Many applications require splines with certain constraints, such as prescribed maximum degree or prescribed order of continuity between the pieces. It is often useful to have a *basis* for the linear vector space of all splines that satisfy such constraints. Besides providing a minimal representation for such splines, the basis often gives valuable insight about the space.

It is relatively easy to compute *some* basis ϕ for a spline space defined in this way. One needs only to set up the linear system that defines the constraints, and solve it by any standard method; the cost of this procedure is usually $O(n^3)$ where n is the size of the mesh. However, the basis elements found by this method are usually nonzero over a large part of the mesh. For efficiency reasons, it is usually desirable to minimize the support of the basis elements. For example, when evaluating a spline f at a point x we need to compute only the values of $\phi_i(x)$ for the elements ϕ_i such that x is in the support of ϕ_i; when computing integrals like $\int \phi_i(x)f(x)dx$ we only need to integrate over the support of ϕ_i. Thus, by reducing the size of the supports we reduce the cost of those computations. For this reason, splines whose support is a small subset of the domain, called *finite elements* (FEs), have become an essential tool in many scientific and engineering disciplines [1].

Finding a finite element basis for a given spline space has been more of an art than a science. There are many specialized constructions that give small (but not necessarity minimal) bases for specific spline spaces, e.g. polynomial splines on triangulations of $\mathbb{R}^2$, $\mathbb{R}^3$ or $\mathbb{S}^2$ with maximum degree g and specified continuity r (C_r). However, there are still may combinations of g and r, and many mesh geometries, for which the optimum basis (or even *any* finite element basis) is not known. There are also many spaces that do not admit any finite-element basis, i.e. for which any basis must include elements whose support is a substantial function of the mesh. However, such a space may still contain a subspace that has a finite element basis, and is large enough for the application at hand. Finding such subspaces, too, is more an art than a science.

For example, consider the space $\mathcal{P}^g_r[\mathcal{C}]$ of trivariate C_r polynomial splines of degree g in a generic tetrahedral partition $\mathcal{C}$ of $\mathbb{R}^3$. According to Lai and Schumaker [2] the problem of finding a basis for $\mathcal{P}^g_r[\mathcal{C}]$ (or just its dimension) seems to be quite difficult unless g is much larger than r. Alfeld *et al.* [3] showed that $\mathcal{P}^g_r[\mathcal{C}]$ has a local basis for $g \geqslant 8r+1$, but they did not give an explicit construction. Alfeld *et al.* [4] gave a construction for $\mathcal{P}^8_1[\mathcal{C}]$. Schumaker and Sorokina [5] stated that they did not know of any general construction for a finite element basis of $\mathcal{P}^5_1[\mathcal{C}]$, but gave an explicit formula for a finite element basis of the subspace of $\mathcal{P}^5_1[\mathcal{C}]$ consisting of all splines which are C_2 on the vertices of $\mathcal{C}$.

For another example, consider a partition $\mathcal{T}$ of $\mathbb{R}^3$ into trihedra with a common vertex at the origin. Let $\mathcal{H}^g_r[\mathcal{T}]/\mathbb{S}^2$ be the space of homogeneous trivariate polynomial splines over $\mathcal{T}$ of degree g, defined on $\mathbb{R}^3$ but restricted to the sphere $\mathbb{S}^2$, with continuity r on $\mathbb{S}^2$. Alfed *et al.* [6] gave an explicit construction for the case $g \geqslant 3r+2$ and conjectured that finite element bases do not exist when $g \leqslant 3r+1$. Gomide and Stolfi [7, 8] described another basis for the space $\mathcal{H}^g_1[\mathcal{T}]/\mathbb{S}^2$ (except for meshes $\mathcal{T}$ with coplanar edges), some of whose elements have smaller support that those given by Alfed *et al.*.

These and many other examples motivated our search for a general algorithm, even if relatively expensive, that would determine a finite element basis with minimum support for an arbitrary spline space $\mathcal{S}$; or, if the space $\mathcal{S}$ does not have such a basis, that would find a large subspace of $\mathcal{S}$ that does. Here we describe such an algorithm [11, 12].

*Institute of Computing, State University of Campinas, {anapaula, stolfi}@ic.unicamp.br

2 Notation and definitions

Meshes and parts. A *mesh over* $\mathbb{R}^n$ is a finite collection of disjoint subsets of $\mathbb{R}^n$, the *parts* of the mesh; such that every part is homeomorphic to a k-dimensional open ball, and there exists an integer d such that every part with dimension $j < d$ is contained in the frontier of a d-dimensional part. The integer d is called the *dimension of* the mesh. A *k-part* is a part with dimension k. The parts of maximum dimension d are called *cells*. The union $\cup\mathcal{C} \subseteq \mathbb{R}^n$ of all parts is the *domain* of $\mathcal{C}$. For lack of space, we will henceforth ignore c-patches where c is not a cell, even though those patches are relevant to the definition of continuity.

Support. The *support* of a spline f on $\mathcal{C}$, denoted by $\operatorname{supp}(f)$, is the set of all cells of $\mathcal{C}$ where f is not identically zero. Note that $\operatorname{supp}(f)$ is a set of cells, not points. The *size* of the support is the number $\#\operatorname{supp}(f)$ of cells in it.

Spline spaces. We will denote by $\langle\phi\rangle$ the linear space generated by a set ϕ of splines. For any subset $\mathcal{K}$ of $\mathcal{C}_d$, we also denote by $\mathcal{S}[\mathcal{K}]$ the subspace of a spline space $\mathcal{S}$ consisting of the splines of $\mathcal{S}$ whose support is contained in $\mathcal{K}$.

Polynomial splines. A *polynomial spline on* a mesh $\mathcal{C}$ over $\mathbb{R}^n$ is a function f defined on the mesh domain $\cup\mathcal{C}$, such that the restriction $f|c$ of f to each part $c \in \mathcal{C}$ (called the *c-patch* of the spline) coincides with some polynomial on the n coordinates of the argument point.

We denote by $\mathcal{P}(\mathcal{C})$ the set of all polynomial splines on the mesh $\mathcal{C}$. Obviously $\mathcal{P}(\mathcal{C})$ is a linear vector space. We also denote by $\mathcal{P}^g_c(\mathcal{C})$ the subspace of $\mathcal{P}(\mathcal{C})$ whose patches have maximum total degree g and are continuous to order c over the entire domain $\cup\mathcal{C}$.

Finite element bases. Let $\mathcal{C}$ be a mesh and $\phi_0, \ldots, \phi_{m-1}$ a basis for some space $\mathcal{S}$ of splines over $\mathcal{C}$. The sum $\sum_{i=0}^{m-1} \#\operatorname{supp}(\phi_i)$ is the *weight* of the basis, denoted by $\operatorname{wt}(\phi)$. Note that the expected cost of evaluating a linear combination $f(x) = \sum_{i=0}^{n-1} a_i\phi_i(x)$ for a random point x is proportional to $\operatorname{wt}(\phi)$.

A *finite element basis* is a basis of splines where $\#\operatorname{supp}(\phi_i)$ is "small" for all i, compared with the total number of mesh elements $\#\mathcal{C}$. The term is meaningful only when applied to *families* of meshes and spline spaces, and it usually means that $\#\operatorname{supp}(\phi_i)$ is limited by a constant that is independent of i and $\#\mathcal{C}$.

In particular, a basis is *piecewise* if the support of each element ϕ_i is a single cell of $\mathcal{C}$. The space $\mathcal{P}^g(\mathcal{C})$ has a piecewise basis, however $\mathcal{P}^g_c(\mathcal{C})$ generaly does not have a piecewise basis when $c \geqslant 0$.

3 The basic algorithm

We describe here a generic algorithm to find a minimum-weight basis for an arbitrary spline space $\mathcal{S}$ on a d-dimensional mesh $\mathcal{C}$. See Algorithm 1.

Algorithm 1

```
1:  $p \leftarrow 0$; $\phi \leftarrow ()$; Set $M^\phi$ to a $0 \times m$ matrix.
2:  $q \leftarrow m$; $\theta \leftarrow \psi$; Set $M^\theta$ to the $m \times m$ identity
    matrix.
3:  $s \leftarrow 1$.
4:  while $p < m$ and $s \leqslant n$ do
5:    for each $\mathcal{K} \subseteq \mathcal{C}_d$ with $\#\mathcal{K} = s$ do
6:      while
7:           there is an element $\xi$ in $\langle\phi, \theta\rangle$ with
               $\operatorname{supp}(\xi) = \mathcal{K}$ that is not in $\langle\phi\rangle$
         do
8:         Append $\xi$ to $\phi$; increment $p$ and adjust $M^\phi$.
9:         Exclude some redundant $\theta_j$ from $\theta$, decre-
           ment $q$ and update $M^\theta$.
10:     end while
11:   end for
12:   $s \leftarrow s + 1$.
13: end while
14: output $\phi$, $M^\phi$.
```

Inputs. The input to Algorithm 1 is an arbitrary basis $\psi_0, \ldots, \psi_{m-1}$ for the space $\mathcal{S}$, and a computable criterion to determine whether a spline is identically zero in a given cell c. Specifically, for each cell $c \in \mathcal{C}_k$ the client must supply a full-rank matrix N^c with r_c rows and m columns, such that, for all i in $0 \ldots r_c - 1$,

$$\sum N^c_{ij} a_j = 0 \Leftrightarrow (\forall x \in c) \sum a_j \psi_j(x) = 0 \qquad (1)$$

For example, we can take $N^c_{ij} = \psi_j(z_i)$ where $\{z_0, z_1, \ldots, z_{r_c-1}\}$ is an appropriate set of points of c. If ψ is a piecewise basis, then N^c is simply the subset of the rows of the identity matrix that correspond to the elements ψ_i whose support is $\{c\}$.

Outputs. The output of the algorithm is another basis $\phi_0, \ldots, \phi_{m-1}$ for $\mathcal{S}$ whose weight $\operatorname{wt}(\phi)$ is minimum among all bases of $\mathcal{S}$. As a byproduct, the algorithm also outputs an $m \times m$ *basis change matrix* M that relates the two bases, that is $\phi_i = \sum_{j=0}^{m-1} M_{ij}\psi_j$.

Invariants. Before each iteration of the inner loop of our algorithm (steps 7–9), we have constructed a partial finite element basis $\phi = (\phi_0, \phi_1, \ldots, \phi_{p-1})$ and a complementary basis $\theta = (\theta_0, \ldots, \theta_{q-1})$, such that $p + q = m$, as well as corresponding basis change matrices, M^ϕ of size $p \times n$ and M^θ of size $q \times n$. These invariants then hold:

P1: $\langle\phi, \theta\rangle = \langle\psi\rangle = \mathcal{S}$.

P2: $\operatorname{wt}(\phi)$ is minimum among all sets of p linearly independent splines of $\mathcal{S}$.

P3: $\phi_i = \sum_{k=0}^{m-1} M^\phi_{ik}\psi_k$ for $i \in 0, \ldots, p-1$.

P4: $\theta_j = \sum_{k=0}^{m-1} M^\theta_{jk}\psi_k$ for $j \in 0, \ldots, q-1$.

At the beginning of each iteration, $\{\theta_0, \ldots, \theta_{q-1}\}$ is a subset of the input basis $\{\psi_0, \ldots, \psi_{m-1}\}$, so the q rows of M^θ are a subset of the rows of $I_{m\times n}$.

Finding a new element. The test in step 7 for the existence of a new element ξ can be performed as follows: (i) determine the subspace $\mathcal{S}[\mathcal{K}]$ of $\mathcal{S} = \langle\phi, \theta\rangle$ that consists of all splines f of $\mathcal{S}$ with $\mathrm{supp}(f) \subseteq \mathcal{K}$, and then (ii) test whether $\mathcal{S}[\mathcal{K}]$ contains any element not in $\langle\phi\rangle$. Since $\mathcal{S}$ has finite dimension, sub-problem (i) can be expressed by a system of linear equations. Solving this system (e. g. by Gaussian elimination) yields a set of r linearly independent splines of $\mathcal{S}$ whose support is contained in $\mathcal{K}$.

If this set includes a spline $\xi = \sum_i u_i\phi_i + \sum_j v_j\theta_j$ with $v_i \neq 0$ for some i, then ξ is not in $\langle\phi\rangle$. Moreover, the support of ξ cannot be strictly contained in $\mathcal{K}$, otherwise it would have been found in step 7 of a previous iteration. Therefore $\mathrm{supp}(\xi) = \mathcal{K}$. Conversely, if all of those splines have $v_0 = v_1 = \cdots = v_{q-1} = 0$, then there is no ξ that satisfies the condition of step 7.

Finding a redundant element. In step 9 we can choose any θ_j such that the expansion of ξ (above) has $v_j \neq 0$. In this step we exclude row j from M^θ, and we insert $(w_0, w_1, \ldots, w_{m-1})$ as row p of M^ϕ, where $w_k = \sum_{i=0}^{p-1} u_i M^\phi_{ik} + \sum_{j=0}^{q-1} v_j M^\theta_{jk}$.

3.1 Correctness

To prove that Algorithm 1 is correct, we need to show that each iteration of steps 7–9 preserves the invariants (P1–P4). Note that this is a “greedy” algorithm [9], that, at each iteration of steps 7–9, adds to the basis ϕ a spline of $\mathcal{S}$ with smallest support that is not yet in $\langle\phi\rangle$. The question is whether greedily adding the smallest possible element ξ at one iteration could somehow prevent us from finding a minimal basis at the end.

Our problem can be represented by a matroid (H, E, K) as defined by Edmonds [10]. The correspondence between Edmonds's notation and ours is as follows: (1) Edmonds's set H of elements of the matroid is our set of all splines of $\mathcal{S}$; (2) an element j of the index set E for Edmonds is for us a coefficient vector a of a spline ξ in terms of the original basis ψ. Therefore, Edmonds's set E is our $\mathbb{R}^m$; (3) Edmonds's weight (or E-weight for short) c_j of that index element is in our algorithm the negative integer $-(\#\,\mathrm{supp}(\sum a_i\psi_i))$; and (4) Edmonds's family K of maximal of independent sets is, in our algorithm, the set of all bases of $\mathcal{S}$.

With these correspondences, our algorithm becomes equivalent to Edmonds's generic greedy algorithm [10, paragraph (7)]. In our case, the E-weight is a negative integer, and the external loop of our algorithm (step 4) considers every possible E-weight $-s$ in decreasing order, and only moves to the next lower E-weight $-(s+1)$ when there are no more basis elements with E-weight equal to $-s$. The “elements already chosen” of Edmonds are the splines $\phi_0, \ldots, \phi_{p-1}$ (more precisely, the coefficients vectors of those splines in terms of the basis ψ). For each s, steps 5 and 7 look for the coefficients $a_0, \ldots, a_{m-1}$ of a spline ξ of $\mathcal{S}$ (i.e. a member j of Edmonds's set E) that is linearly independent of the splines $\phi_0, \ldots, \phi_{p-1}$. Therefore, Algorithm 1 is an instance of Edmonds's, and his proof of correctness [10, paragraphs 18–28] holds for our algorithm too. □

3.2 Efficiency

Algorithm 1 has exponential running time since steps 6–9 are executed 2^n times in the worst case, but it can be improved in many ways. As we shall see, for most cases of interest its running time can be reduced to polynomial — and eventually linear — in the size of the mesh.

Note however that the algorithm stops as soon as $p = m$, since step 7 will then certainly fail for all $\mathcal{K}$. Thus, if $\mathcal{S}$ has a basis whose maximum support size is t, the algorithm performs only $\binom{n}{0} + \cdots + \binom{n}{t} + t$ iterations of steps 7–9, which is $O(n^t)$. Since the cost of one iteration of steps 7–9 is $O(m^3)$, the total time will be $O(n^t m^3)$.

3.3 Exploiting connectedness

We can improve the efficiency even further by observing that some sets $\mathcal{K}$ cannot possibly provide a new element ξ. A subset $\mathcal{K} \subseteq \mathcal{C}_d$ is *connected* with respect to a spline space $\mathcal{S}$ if for every non-trivial partition $\mathcal{K}_1$, $\mathcal{K}_2$ of $\mathcal{K}$ we have $\mathcal{S}[\mathcal{K}] \neq \mathcal{S}[\mathcal{K}_1] \oplus \mathcal{S}[\mathcal{K}_2]$.

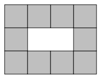
Theorem 1 *In a basis ϕ of minimum weight for a spline space over a mesh $\mathcal{C}$, the support of each element ϕ_i is a connected set of cells of $\mathcal{C}_d$.*

For the proof of this theorem and the details on how to test a set $\mathcal{K}$ for connectedness in this sense, see the full version of this paper [12].

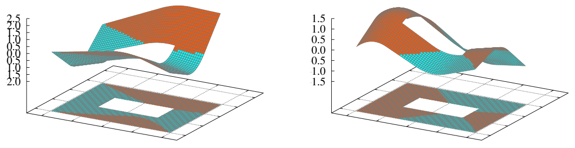
With theorem 1, we can speed up Algorithm 1 by considering only subsets $\mathcal{K} \subseteq \mathcal{C}_k$ that are connected in the graph G. Namely we replace step 5 of the algorithm by: “**for** each $\mathcal{K} \subseteq \mathcal{C}_d$ with $\#\mathcal{K} = s$ such that $G[\mathcal{K}]$ is connected **do**”.

For many meshes of practical interest, there is a relatively small bound h on the number of neighbors of each cell, independent of the total number n of cells. Moreover the constraints are usually continuity requirements that relate coefficients $a_{j'}$, $a_{j''}$ which are in adjacent cells. Therefore the maximum vertex degree of the graph G is h, and the number of connected subgraphs of G with s nodes is $O(h^s n)$. It follows that the cost of each iteration of steps 7–9 is $O(h^s n)$. So, the total time will be $O(h^s n m^3)$, where s is the maximum support size of any element in the minimum weight basis. Alternatively, the algorithm can be stopped after s reaches a preset maximum support size, in which case it will return a basis of minimum weight in the largest subspace of $\mathcal{S}$ which has a basis whose element supports do not exceed s.

4 Example

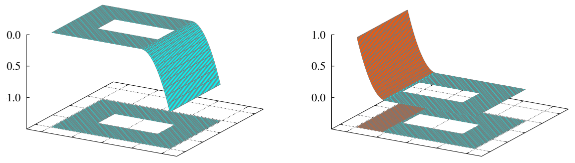


Below we show the basis found with algorithm 1 for the space $\mathcal{S} = \mathcal{P}^2_1[\mathcal{C}]$ on the 10-cell mesh $\mathcal{C}$, shown at left. For the initial basis ψ, we used a set of linearly independent splines in $\mathcal{P}^2_1[\mathcal{C}]$, derived from a piecewise basis of $\mathcal{P}^2[\mathcal{C}]$ by solving the C_2 continuity constraints. The space has dimension 11. Two of these input splines are shown in figure 1. Figure 2 shows four elements of the basis ϕ found by Algorithm 1. Note that the support of ϕ_{11} is the whole mesh. This is unavoidable since the space $\mathcal{P}^2_1[\mathcal{C}]$ does not admit a finite-element basis.

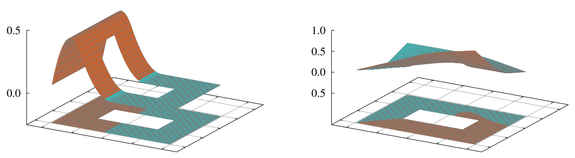


ψ_3 (wt $= 10$) ψ_{10} (wt $= 10$)

Figure 1: Two elements of the input basis ψ for the space $\mathcal{P}^2_1[\mathcal{C}]$.



ϕ_1 (wt $= 3$) ϕ_2 (wt $= 3$)



ϕ_6 (wt $= 5$) ϕ_{11} (wt $= 10$)

Figure 2: Some elements of a minimum-weight basis ϕ for the space $\mathcal{P}^2_1[\mathcal{C}]$.

The C program and data files for these tests are available at `http://www.ic.unicamp.br/~anapaula/minimalbases.tar.gz`.

5 Conclusions

We have described an algorithm that finds a finite element basis with minimal weight in an arbitrary spline space. Alternatively, the algorithm can be used to find a maximal subspace of a given space $\mathcal{S}$ that admits a basis whose elements have a prescribed maximum support size t. In either case, the cost grows exponentially on s, but nevertheless the algorithm is viable for many meshes and spaces of practical importance.

Acknowledgment

The authors acknowledge support of FAPESP (grant 2007/52015-0) and CNPq (grants 143088/05-0, 306631/07-5, 551233/09-5).

References

[1] R. H. Gallagher, *Finite Element Analysis: Fundamentals*. Prentice-Hall, 1975.

[2] M.-J. Lai and L. L. Schumaker, "Trivariate C^r polynomial macroelements," *Constructive Approximation*, vol. 26, no. 1, pp. 11–28, 2007.

[3] P. Alfeld, L. L. Schumaker, and M. Sirvent, "On dimension and existence of local bases for multivariate spline spaces," *J. Approx. Theory*, vol. 70, pp. 243–264, 1992.

[4] P. Alfeld, L. L. Schumaker, and W. Whiteley, "The generic dimension of the space of C^r splines of degree $d \geq 8$ on tetrahedral decompositions," *SIAM J. Numer. Anal.*, vol. 30, no. 3, pp. 889–920, 1993.

[5] L. L. Schumaker and T. Sorokina, "C^1 quintic splines on type-4 tetrahedral partitions." *Adv. Comput. Math.*, vol. 21, no. 3–4, pp. 421–444, 2004.

[6] P. Alfed, M. Neamtu, and L. L. Schumaker, "Dimension and local bases of homogeneous spline spaces," *SIAM Journal of Mathematical Analysis*, vol. 27, no. 5, pp. 1482–1501, September 1996.

[7] A. Gomide and J. Stolfi, "Bases for Non-Homogeneous Polynomial $\mathbf{C}_k$ Splines on the Sphere," *Proc. LATIN'98 - Latin American Theoretical Informatics Conference*, pp. 133–140, 1998.

[8] A. Gomide, "Splines polinomiais não homogêneos na esfera," Ph.D. dissertation, Institute of Computing, University of Campinas, May 1999.

[9] Herbert S. Wilf, *Algorithms and Complexity*. Prentice-Hall, 1986.

[10] J. Edmonds, "Matroids and the greedy algorithm," *Mathematical Programming*, vol. 1, pp. 127–136, 1971.

[11] Ana Paula Resende Malheiro and Jorge Stolfi. "Finding minimal bases in arbitrary spline spaces." *Technical Report IC-09-21*, Institute of Computing, University of Campinas, June 2009.

[12] Ana Paula Resende Malheiro and Jorge Stolfi. "Finding minimal bases in arbitrary spline spaces (full paper)," *Proc. 2010 Canadian Conference on Computational Geometry* (CD-ROM).