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. 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. When working with such splines, it is 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; and, 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 [2].

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 [5] 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, Schumaker and Sirvent [6] 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, Schumaker and Whiteley [7] gave an explicit construction for $\mathcal{P}^8_1[\mathcal{C}]$. Schumaker and Sorokina [8] 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 $\mathcal{P}^5_1[\mathcal{C}]$ whose splines are C_2 on the vertices of $\mathcal{C}$. Hecklin, Nürnberger, Schumaker and Zeilfelder [9] constructed a finite element basis for $\mathcal{P}^3_1[\mathcal{C}]$ where $\mathcal{C}$ is a specific tetrahedral mesh derived from a uniform cubical mesh in $\mathbb{R}^3$.

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, Neamtu and Schumaker [10] 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 [11, 12] 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.

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

2 Notation and definitions

Before we describe our algorithm, we need to define some basic concepts and notation. See table 1.

2.1 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. For this work the precise nature of the meshes is not important, as long as 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; we denote by $\mathcal{C}_k$ the *k-skeleton* of $\mathcal{C}$, that is, the subset of $\mathcal{C}$ consisting of all its k-parts. The 0-parts and 1-parts are called *vertices* and *edges*, respectively. 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 simplicity, we will assume that the value of a spline on any point of $\cup\mathcal{C}$ that is not inside a cell — that is, a point on the k-skeleton of $\mathcal{C}$, with $k < d$ — is some fixed convex combination of the limiting values of the spline in the adjacent cells. In particular, if the d-dimensional pieces of a spline are continuous across the k-part, the spline will be continuous also over that k-part; and a spline will be identically zero over $\cup\mathcal{C}$ if and only if it is zero over $\cup\mathcal{C}_d$. This assumption allows us to ignore the lower-dimensional pieces of the spline in the remainder of the article.

2.2 Support

The *support* of a spline f on $\mathcal{C}$, denoted by $\mathrm{supp}(f)$, is the set of all cells of $\mathcal{C}$ where f is not identically zero. Note that $\mathrm{supp}(f)$ is a set of cells, not points; so that $\cup\,\mathrm{supp}(f)$ may be larger than the set of *points* in the cells where f is different of zero. The *size* of the support is the number $\#\,\mathrm{supp}(f)$ of cells in it.

2.3 Spline spaces and subspaces

We will denote by $\langle\phi\rangle$ the linear space generated by a set ϕ of splines over the same mesh $\mathcal{C}$; that is, the set of all splines f such that

$$f = \sum_{i=0}^{m-1} a_i\phi_i \tag{1}$$

for some coefficients $a_0, \ldots, a_{n-1} \in \mathbb{R}$.

Any finite-dimensional space $\mathcal{S}$ of polynomial splines has a finite *basis*, that is, a list $\phi = (\phi_0, \ldots, \phi_{m-1})$ of linearly independent splines of $\mathcal{S}$ such that $\langle\phi\rangle = \mathcal{S}$.

For any subset $\mathcal{K}$ of $\mathcal{C}$, and any spline space $\mathcal{S}$ we denote by $\mathcal{S}[\mathcal{K}]$ the subspace of $\mathcal{S}$ consisting of the splines of $\mathcal{S}$ whose support is contained in $\mathcal{K}$.

Symbols	Meaning	Section
$\mathcal{C}$	set of cells of the mesh	2.1
n	number of cells	5
d	dimension of the mesh	2.1
$\mathcal{P}(\mathcal{C})$	space of all polynomial splines on $\mathcal{C}$	2.3
g	degree	2.4
r	continuity order	2.4
$\mathcal{P}^g_r(\mathcal{C})$	splines of $\mathcal{P}(\mathcal{C})$ with degree g and continuity r	2.4
$\mathcal{S}, \mathcal{S}_1, \mathcal{S}_2$	spline spaces	2.3
m	dimension of a spline space	5
$\mathcal{K}, \mathcal{K}_1, \mathcal{K}_2$	subsets of cells	2.3
ϕ, ψ, ξ	spline bases	2.3
ϕ_i, ψ_i	elements of a basis	2.3
ξ^c_k	basis element k associated with cell c	7
$\langle\phi\rangle$	space generated by splines in ϕ	2.3
$\#X$	cardinality of set X	2.4
$\mathrm{supp}(f)$	set of cells where f is nonzero	2.4
$\mathrm{wt}(\phi)$	weight of basis, $\sum_i \#\,\mathrm{supp}(\phi_i)$	3

Table 1: Index of symbols

2.4 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 will denote by $\mathcal{P}(\mathcal{C})$ the set of all polynomial splines on the mesh $\mathcal{C}$. It is easy to see that $\mathcal{P}(\mathcal{C})$ is a linear vector space.

If the dimension d of $\mathcal{C}$ is positive, the space $\mathcal{P}(\mathcal{C})$ has infinite dimension. However, if we specify a maximum degree g for the polynomials that define the patches, we get a finite-dimensional subspace $\mathcal{P}^g(\mathcal{C})$ of $\mathcal{P}(\mathcal{C})$. If we specify additional linear constraints on the splines (for example, continuity constraints between adjacent patches), we get various linear subspaces of $\mathcal{P}^g(\mathcal{C})$. An important example is the space $\mathcal{P}^g_r(\mathcal{C})$ of all splines of $\mathcal{P}^g(\mathcal{C})$ that are continuous to order r over the entire domain $\cup\mathcal{C}$.

3 Finite element bases

Let $\phi_0, \phi_1, \ldots, \phi_{m-1}$ be a basis for a space $\mathcal{S}$ of splines over same mesh $\mathcal{C}$. The sum $\sum_{i=0}^{m-1} \#\operatorname{supp}(\phi_i)$ is the *weight* of the basis, denoted by $\operatorname{wt}(\phi)$.

The weight of a basis ϕ is related to its computational efficiency. Suppose that we can efficiently identify the cell c of $\mathcal{C}$ that contains a given point $x \in \mathbb{R}^n$, and obtain the list of all basis elements ϕ_i that are nonzero in c. The cost of computing $f(x)$ by formula (1) is then the number of those elements times the mean cost of evaluating one piece of each ϕ_i. Suppose now that x is a random point of $\cup\mathcal{C}$, such that (1) the probability that x belongs to a cell $c \in \mathcal{C}$ is the same for all the cells, and (2) the probability that x belongs to any j-part with $j < d$ is zero. It is easy to see that the expected cost of computing $f(x)$ by formula (1) is essentially the cost of evaluating one piece of each ϕ_i times $\sum_{i=0}^{m-1} \#\operatorname{supp}(\phi_i)$, that is, times the weight $\operatorname{wt}(\phi)$. Therefore the expected evaluation cost of formula (1) is minimum when $\operatorname{wt}(\phi)$ is minimum.

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 $\mathcal{P}^g(\mathcal{C})$ (without any continuity contraints) has infinitely many (finite) piecewise bases. One may take, for example, the canonical basis for the d-variate polynomials (namely, all monomials of degree $\leqslant g$ in d variables) and restrict each of its elements to each part of $\mathcal{C}$. For meshes consisting of triangles, one may take instead the Bernstein-Bezier polynomials on the barycentric coordinates of each cell. However $\mathcal{P}^g_c(\mathcal{C})$ generaly does not have a piecewise basis when $c \geqslant 0$.

4 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 $\mathcal{C}$. The main procedure is Algorithm 1 below, which is explained in the rest of this section, and improved in the following sections.

Algorithm 1

$p \leftarrow 0$; $\phi \leftarrow ()$; Set M^ϕ to a $0 \times m$ matrix.
$q \leftarrow m$; $\theta \leftarrow \psi$; Set M^θ to the $m \times m$ identity matrix.
for $s = 1, \ldots, n$ **do**
  for every $\mathcal{K} \subseteq \mathcal{C}_k$ such that $\#\mathcal{K} = s$ **do**
    while
      there is an element ξ in $\langle\phi, \theta\rangle$ with $\operatorname{supp}(\xi) = \mathcal{K}$ that is not in $\langle\phi\rangle$
    do
      append ξ to ϕ; increment p and adjust M^ϕ;
      exclude some redundant θ_j from θ; decrement q and update M^θ;
    end while
  end for
end for
output ϕ, M^ϕ.

4.1 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 $0 \ldots r_c - 1$,

$$\sum N^c_{ij} a_j = 0 \quad \Leftrightarrow \quad (\forall x \in c) \sum a_j \psi_j(x) = 0 \qquad (2)$$

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 $\{c\}$.

4.2 Outputs

The output of the algorithm is another basis $\phi_0, \ldots, \phi_{m-1}$ for $\mathcal{S}$ whose weight 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:

$$\phi_i = \sum_{j=0}^{m-1} M_{ij}\psi_j \quad (3)$$

4.3 Description of the algorithm

Invariants. Before each iteration of the inner loop of our algorithm (steps 6–8), 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 $q \times n$. The invariants then hold:

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

P2: $\mathrm{wt}(\phi)$ 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 the $m \times m$ identity matrix.

Finding the new element. The test of step 6 can be performed as follows: (a) determine the subspace $\mathcal{S}[\mathcal{K}]$ of $\mathcal{S} = \langle \phi, \theta \rangle$ that consists of all splines f with $\mathrm{supp}\, f \subseteq \mathcal{K}$, and then (b) test whether $\mathcal{S}[\mathcal{K}]$ contains any element not in $\langle \phi \rangle$. Since $\mathcal{S}$ has finite dimension, item (a) means solving a system of linear equations. Therefore, to perform tests (a) and (b) above, we build the system

$$N^\mathcal{K} M^{-1} a = 0 \quad (4)$$

where

- $N^\mathcal{K}$ is the vertical concatenation of the matrices N^c for all $c \in \overline{\mathcal{K}}$;
- M is the current basis change matrix, the vertical concatenation of M^ϕ and M^θ; and
- a is a vector with m coefficients, the concatenation of p coefficients $(u_0, \ldots, u_{p-1})$ for ϕ and q coefficients $(v_0, \ldots, v_{q-1})$ for θ.

To ensure condition (b) we add to this system the equation

$$u_i = 0 \quad (5)$$

for every i such that $\mathrm{supp}_d\, \phi_i \subseteq \mathcal{K}$.

Solving this system [15] yields a set of r linearly independent vectors $(u_0, \ldots, u_{p-1}, v_0, \ldots, v_{q-1})$ that satisfy system (4) and (5); that is, r linearly independent splines of $\mathcal{S}$ whose support is contained in $\mathcal{K}$.

If one of these vectors has $v_i \neq 0$ for some i, then the corresponding spline $\xi = \sum_i u_i\phi_i + \sum_j v_j\theta_j$ is not in $\langle \phi \rangle$. Moreover, the support of ϕ cannot be strictly contained in $\mathcal{K}$, otherwise it would have been found in a precious iteration of steps 6 through 9. Therefore $\mathrm{supp}(\xi) = \mathcal{K}$. Conversely, if all of those vectors have $v_0 = v_1 = \cdots = v_{q-1} = 0$ then all the splines that satisfy system (4) are in $\langle \phi \rangle$, and there is no ξ that satisfies the condition of step 6.

Finding a redundant element. The new element ξ found in step 6 can be written as $\xi = \sum_{i=0}^{p-1} u_i\phi_i + \sum_{j=0}^{q-1} v_j\theta_j$. In step 8 we can choose any θ_j such that $v_j \neq 0$. 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 $w_k = \sum_{i=0}^{p-1} u_i M^\phi_{ik} + \sum_{j=0}^{q-1} v_j M^\theta_{jk}$.

4.4 Correctness

To prove that Algorithm 1 is correct, we need to show that each iteration of steps 6–8 preserves the invariants (P1–P4). Note that this is a "greedy" algorithm [13], that, at each iteration of steps 6–8, adds to a basis ϕ the 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 [14]. Here is the correspondence between Edmonds's notation and ours:

- Edmonds's set H of elements of the matroid is our set of all splines of $\mathcal{S}$;
- 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. Thus, Edmonds's set E is our $\mathbb{R}^m$;
- 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
- 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 [14, paragraph (7)]:

> *in each step, choose any largest weight member of E, not already chosen, which together with the members already chosen forms a subset of some member of K, and stop when the chosen members of E comprise a member of K.*

In our problem, the E-weights are negative integers. The external loop of our algorithm (step 3), 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 $-s$. For each s, steps 4 through 6 look for the coefficients $a_1, \ldots, a_m$ of a spline of $\mathcal{S}$ (i.e. a member j of E) that is linearly independent of the splines $\phi_0, \ldots, \phi_{p-1}$ already chosen. The "elements already chosen" are the splines $\phi_1, \ldots, \phi_p$ (more precisely the coefficients vectors of those splines in terms of the basis ψ). Thus, the correctness of the algorithm is proved by Edmonds [14, paragraphs (18 – 28)]. □

5 Efficiency

The efficiency of this algorithm depends on how many times the test of step 6 is performed.

The two outer **for** loops of Algorithm 1 enumerate all 2^n subsets $\mathcal{K}$ of $\mathcal{C}_d$, where n is the number of cells in the mesh, in order of increasing cardinality. For each iteration of the **for** loops, the test of the **while** loop is executed $t_{\mathcal{K}}+1$ times, where $t_{\mathcal{K}}$ is the number of elements ξ found for that set $\mathcal{K}$. Since the sum of all $t_{\mathcal{K}}$ is m, the dimension of the space $\langle\psi\rangle$, the algorithm runs in time $(2^n+m)T$ where T is the time to build and solve system (4) – which is $O(m^3)$.

6 Optimizations

Algorithm 1 can be improved in many ways. As we shall see, for most cases of interest its running time can be reduced from exponential to polynomial — and eventually linear — in the size of the mesh.

6.1 Early stopping

For one thing, we can stop as soon as $p = m$, since step 3 will then certainly fail for all $\mathcal{K}$. Thus, if $\mathcal{S}$ has a basis whose maximum support size is t, the algorithm runs in only $\binom{n}{0} + \cdots + \binom{n}{t} + t$ iterations of step 6, which is $O(n^t)$. Since the cost of one iteration of steps 6 … 8 is $O(m^3)$, the total time will be $O(n^t m^3)$.

6.2 Exploiting connectivity

We can improve the efficiency even further by observing that some sets $\mathcal{K}$ cannot possibly provide a new element ξ. A subset $\mathcal{K} \subseteq \mathcal{C}$ 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

$$\mathcal{S}[\mathcal{K}] \neq \mathcal{S}[\mathcal{K}_1] \oplus \mathcal{S}[\mathcal{K}_2] \tag{6}$$

Theorem 1 *In a basis of minimum weight, the support of each element ϕ_i is a connected set of cells of $\mathcal{C}$.*

Proof. Let ϕ a basis of minimum weight for a space $\mathcal{S}$. Suppose for contradiction that $\operatorname{supp}\phi_i$ is not connected, that is, $\operatorname{supp}\phi_i$ is a set $\mathcal{K} = \mathcal{K}_1 \uplus \mathcal{K}_2$ satistying (6). Then ϕ_i can be written as $\phi' + \phi''$ where $\phi' \in \mathcal{S}[\mathcal{K}_1]$ and $\phi'' \in \mathcal{S}[\mathcal{K}_2]$. Therefore, if we remove ϕ_i and add ϕ' and ϕ'', the resulting set still generates the space $\mathcal{S}$. Since this substitution increases the number of elements by one, there must be some linear dependence between the elements of the resulting basis, that is, there is an element ϕ^* that is a linear combination of ϕ' and/or ϕ'' and/or other elements ϕ_j.

If we exclude this element ϕ^*, we are left with a basis for $\mathcal{S}$ whose total weight is $\operatorname{wt}(\phi) - \#\operatorname{supp}(\phi_i) + \#\operatorname{supp}(\phi') + \#\operatorname{supp}(\phi'') - \#\operatorname{supp}(\phi^*) = \operatorname{wt}(\phi) - \#\operatorname{supp}(\phi^*) < \operatorname{wt}(\phi)$ contradicting the hypothesis that ϕ had minimum weight. □ □

6.3 Finding connected subsets of cells

Suppose that the space $\mathcal{S}$ is defined in terms of a piecewise basis β of size t (such that $\#\operatorname{supp}(\beta_i) = 1$ for every i) by a set of r homogeneous linear constraints

$$s \in \mathcal{S} \Leftrightarrow \sum_{j=0}^{t-1} R_{ij} a_j = 0,\ i = 0, 1, \ldots, r-1 \tag{7}$$

where $a_0, \ldots, a_{t-1}$ are the coeficients of the spline s relative to β, and R is an $r \times t$ matrix. These constraints may be continuity conditions between adjacent cells, boundary conditions, etc.

Consider the graph G derived from the matrix R as follows. Each vertex of G is a cell of $\mathcal{C}$, and there is an edge between two vertices c', $c'' \in \mathcal{C}$ iff there is some equation that relates the coefficients of those two cells, that is, a row of R which has two nonzero elements $R_{ij'}$ and $R_{ij''}$, where $\operatorname{supp}(\beta_{j'}) = \{c'\}$ and $\operatorname{supp}(\beta_{j''}) = \{c''\}$.

Theorem 2 *For any $\mathcal{K} \subseteq \mathcal{C}$, if the induced graph $G[\mathcal{K}]$ is disconnected, then $\mathcal{K}$ is disconnected relative to the spline space $\mathcal{S}$.*

Proof. Suppose the graph $G[\mathcal{K}]$ is disconnected, that is, there are sets $\mathcal{K}_1, \mathcal{K}_2$ such that $\mathcal{K} = \mathcal{K}_1 \cup \mathcal{K}_2$, $\mathcal{K}_1 \neq \emptyset$, $\mathcal{K}_2 \neq \emptyset$, and there is no edge of G between $\mathcal{K}_1$ and $\mathcal{K}_2$. Then we can rearrange the rows of matrix R and the basis elements β so that

$$R = \begin{bmatrix} A_1 & 0 & B_1 \\ 0 & A_2 & B_2 \\ 0 & 0 & A_3 \end{bmatrix} \tag{8}$$

where A_1 and B_1 represent the equations that involve a cell of $\mathcal{K}_1$; A_2 and B_2 those that involve a cell of $\mathcal{K}_2$; and A_3 those that do not involve any cell of $\mathcal{K}$.

In the subspace $\mathcal{S}[\mathcal{K}]$, all coefficients a_j such that $\operatorname{supp}(\beta_j) \notin \mathcal{K}$ are zero. Therefore, we can describe $\mathcal{S}[\mathcal{K}]$ by a set of equations.

$$\begin{bmatrix} A_1 & 0 & 0 \\ 0 & A_2 & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{9}$$

where the sub-vectors a_1 and a_2 are the coefficients corresponding to elements β_i in $\mathcal{K}_1$ and $\mathcal{K}_2$, respectively, and a_3 are the coefficients corresponding to elements in $\overline{\mathcal{K}}$. Similarly, the splines of $\mathcal{S}[\mathcal{K}_1]$ are the solutions of:

$$\begin{bmatrix} A_1 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad (10)$$

and, the splines of $\mathcal{S}[\mathcal{K}_2]$ are defined by:

$$\begin{bmatrix} I & 0 & 0 \\ 0 & A_2 & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad (11)$$

It follows that an arbitrary spline of $\mathcal{S}[\mathcal{K}]$ is an arbitrary spline of $\mathcal{S}[\mathcal{K}_1]$ added to an arbitrary spline of $\mathcal{S}[\mathcal{K}_2]$, that is, $\mathcal{S}[\mathcal{K}] = \mathcal{S}[\mathcal{K}_1] \oplus \mathcal{S}[\mathcal{K}_2]$. □ □

6.4 Number of connected subsets

In light of Theorem 2, we can speed up Algorithm 1 by considering only subsets $\mathcal{K} \subseteq \mathcal{C}_k$ that are connected in the graph G. This version is shown in Algorithm 2.

Algorithm 2

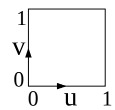
1: $p \leftarrow 0$; $\phi \leftarrow ()$; set M^ϕ to a $0 \times m$ matrix.
2: $q \leftarrow m$; $\theta \leftarrow \psi$; set M^θ to the $m \times m$ identity matrix.
3: $s \leftarrow 1$
4: **while** $p < m$ and $s \leqslant n$ **do**
5: **for** each connected subset $\mathcal{K} \subseteq \mathcal{C}_k$ of G with $\#\mathcal{K} = s$ **do**
6: **while**
7: there is an element ξ in $\langle \phi, \theta \rangle$ with $\mathrm{supp}_d\, \xi = \mathcal{K}$ that is not in $\langle \phi \rangle$
8: **do**
9: append ξ to ϕ; increment p and adjust M^ϕ;
10: exclude some redundant θ_j from θ; decrement q and update M^θ;
11: **end while**
12: **end for**
13: $s \leftarrow s + 1$
14: **end while**

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 C are usually continuity constraints 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 iteration of steps 7 – 10 is $O(h^s n)$. Therefore, 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, Algorithm 2 can be used to find the basis of minimum weight in the space $\mathcal{S}$ whose element supports do not exceed a specified size s.

7 Examples

In this section we show three examples with meshes that are subsets of the unit regular square grid. We consider the piecewise basis β which, in each cell c, has the following elements:



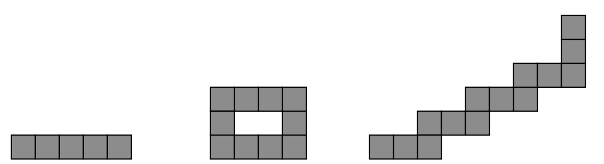
$$\beta_0^c = (1-u)(1-v) \qquad \beta_3^c = uv$$
$$\beta_1^c = u(1-v) \qquad \beta_4^c = u(1-u)$$
$$\beta_2^c = (1-u)v \qquad \beta_5^c = v(1-v)$$

where u and v are cell-relative coordinates as in the figure at left. This basis generates the space $\mathcal{P}^2[\mathcal{C}]$ of all splines of total degree 2 (not necessarily continuous) over the mesh $\mathcal{C}$. Therefore, if $\mathcal{C}$ is a mesh with n cells, each spline of $\mathcal{P}^2[\mathcal{C}]$ is defined by $6n$ coefficients a_i^c where $c \in \mathcal{C}$ and $i \in 0 \ldots 5$.

In all three these examples we consider the subspace $\mathcal{S} = \mathcal{P}_1^2[\mathcal{C}]$ of $\mathcal{P}^2[\mathcal{C}]$ that consists of continuous splines with continous 1st derivatives. A spline in $\mathcal{P}_1^2[\mathcal{C}]$ is defined by the following $\mathcal{C}^1$ *continuity constraints* between every two horizontally adjacent cells c', c'':

$$\begin{cases} a_1^{c'} - a_0^{c''} = 0 \\ a_3^{c'} - a_2^{c''} = 0 \\ a_5^{c'} - a_5^{c''} = 0 \\ -a_0^{c'} + a_1^{c'} - a_4^{c'} + a_0^{c''} - a_1^{c''} - a_4^{c''} = 0 \\ a_0^{c'} - a_1^{c'} - a_2^{c'} + a_3^{c'} - a_0^{c''} + a_1^{c''} + a_2^{c''} - a_3^{c''} = 0 \end{cases} \qquad (12)$$

The first three equations are $\mathcal{C}^0$ continuity constraints while the last two impose the continuity of derivatives, assuming that the $\mathcal{C}^0$ constraints are met. Similar equations hold for vertically adjacent cells. The meshes $\mathcal{C}$ used in the examples are shown in figure 1.



(1) (2) (3)

Figure 1: The meshes $\mathcal{C}$ used in examples 1, 2 e 3.

Table 2 summarizes these tests, where $n = \#\mathcal{C}_d$ is the number of cells, m is the dimension of $\mathcal{P}_1^2[\mathcal{C}]$, ψ is the starting basis for $\mathcal{S}$, and ϕ is the optimal basis.
Figure 2 shows the input basis ψ for the example mesh 1, and figure 3 is the minimum-weight basis ϕ found by Algorithm 2. Figure 4 is input basis ψ for mesh 2, and figure 5 the corresponding output basis ϕ. Figure 6

Mesh	n	m	wt(ψ)	wt(ϕ)
1	5	10	46	30
2	14	19	204	84
3	10	11	101	60

Table 2: Summary of the examples.

shows the minimum-weight basis ϕ for the mesh 3. In all examples, note that the support of one or more elements at the end of the basis $\phi_{m-k}, \phi_{m-k+1}, \ldots, \phi_{m-1}$ is the whole mesh. This is unavoidable since the space $\mathcal{P}_1^2[\mathcal{C}]$ does not admit a finite-element basis. Nevertheless, if those k elements are excluded, the remaining elements $\phi_0, \ldots, \phi_{m-1-k}$ are a minimal finite-element basis for the subspace that they generate; which is the largest subspace of $\mathcal{P}_1^2[\mathcal{C}]$ that admits a base without whole-mesh elements.

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

8 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] J. H. Argyris and S. Kelsey, "Energy theorems and structural analysis," *Butterworth Scientific*, 1960.

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

[3] M. J. Turner, R. W. Clough, H. C. Martin, and L. P. Topp, "Stiffness and deflection analysis of complex structures," *J. Aeronautical Society*, vol. 23, 1956.

[4] O. C. Zienkiewicz, "The finite element method: From intuition to generality," *Applied Mechanics Review*, vol. 23, no. 3, pp. 249–256, 1970.

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

[6] 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.

[7] P. Alfeld, L. L. Schumaker, and W. Whiteley, "The generic dimension of the space of C^1 splines of degree $d \geq 8$ on tetrahedral decompositions," *SIAM J. Numer. Anal.*, vol. 30, no. 3, pp. 889–920, 1993.

[8] 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.

[9] G. Hecklin, G. Nürnberger, L. Schumaker, and F. Zeilfelder, "A local lagrange interpolation method based on C^1 cubic splines on freudenthal partitions," *Math. Comp*, vol. 77, no. 262, pp. 1017–1036, 2007.

[10] 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.

[11] 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.

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

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

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

[15] 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.

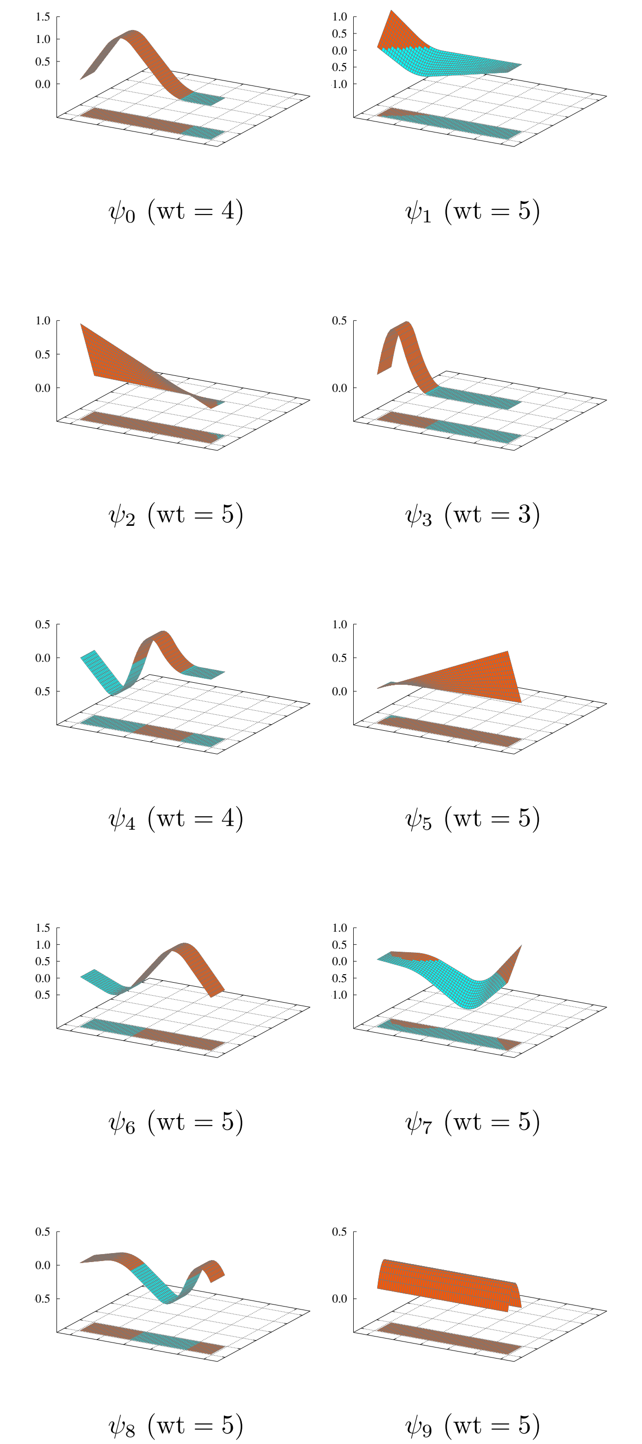


Figure 2: The input basis ψ for the space $\mathcal{P}_1^2[\mathcal{C}]$, where $\mathcal{C}$ is the mesh (1) of figure 1.

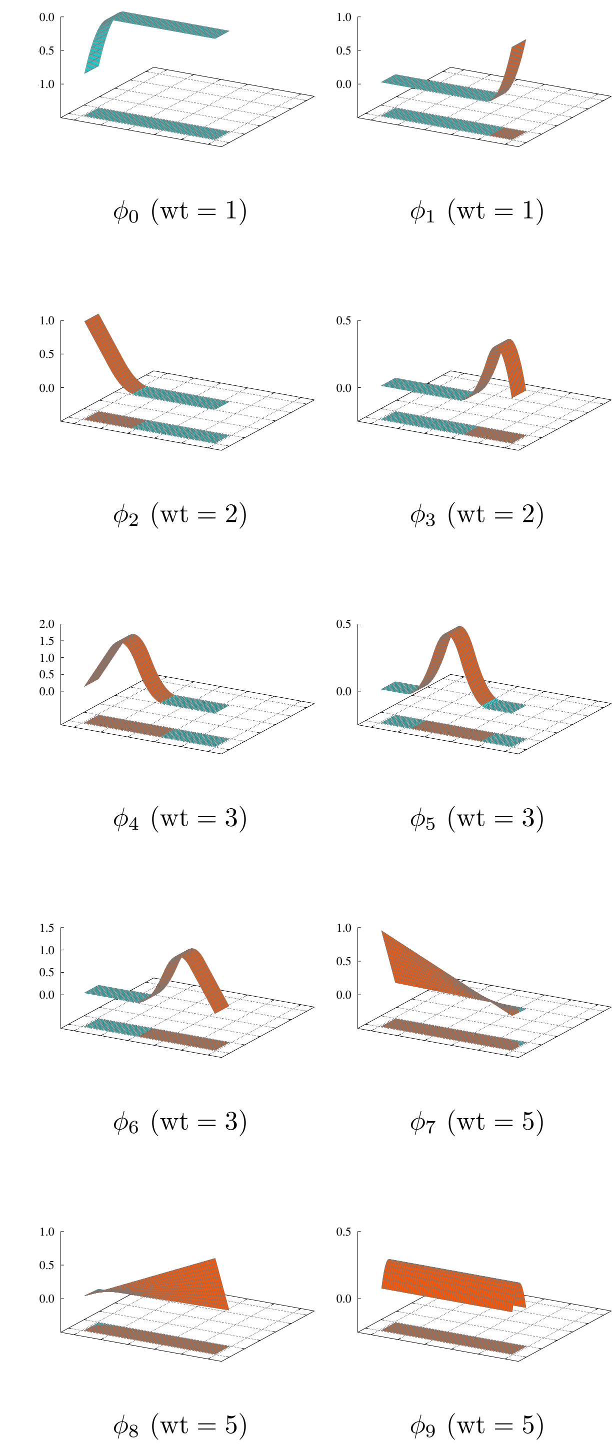


Figure 3: A minimum-weight basis ϕ for the space $\mathcal{P}_1^2[\mathcal{C}]$ where $\mathcal{C}$ is the mesh (1) of figure 1.

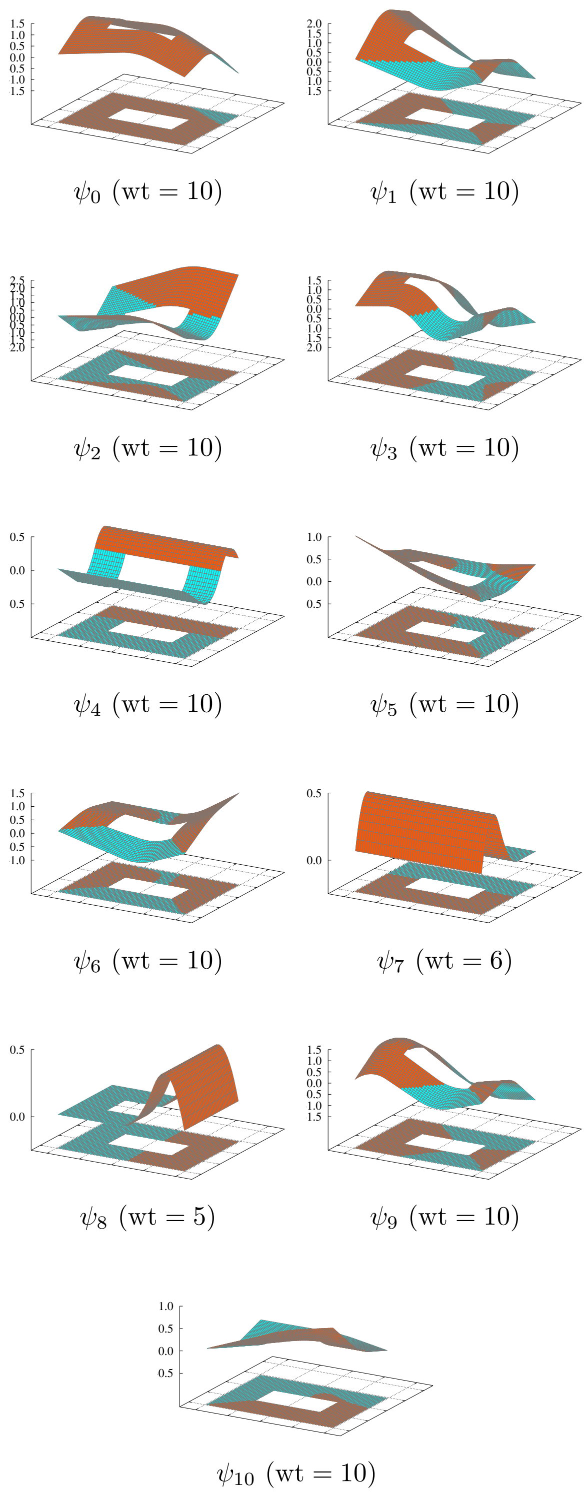


Figure 4: The input basis ψ for the space $\mathcal{P}_1^2[\mathcal{C}]$, where $\mathcal{C}$ is the mesh (2) in the figure 1.

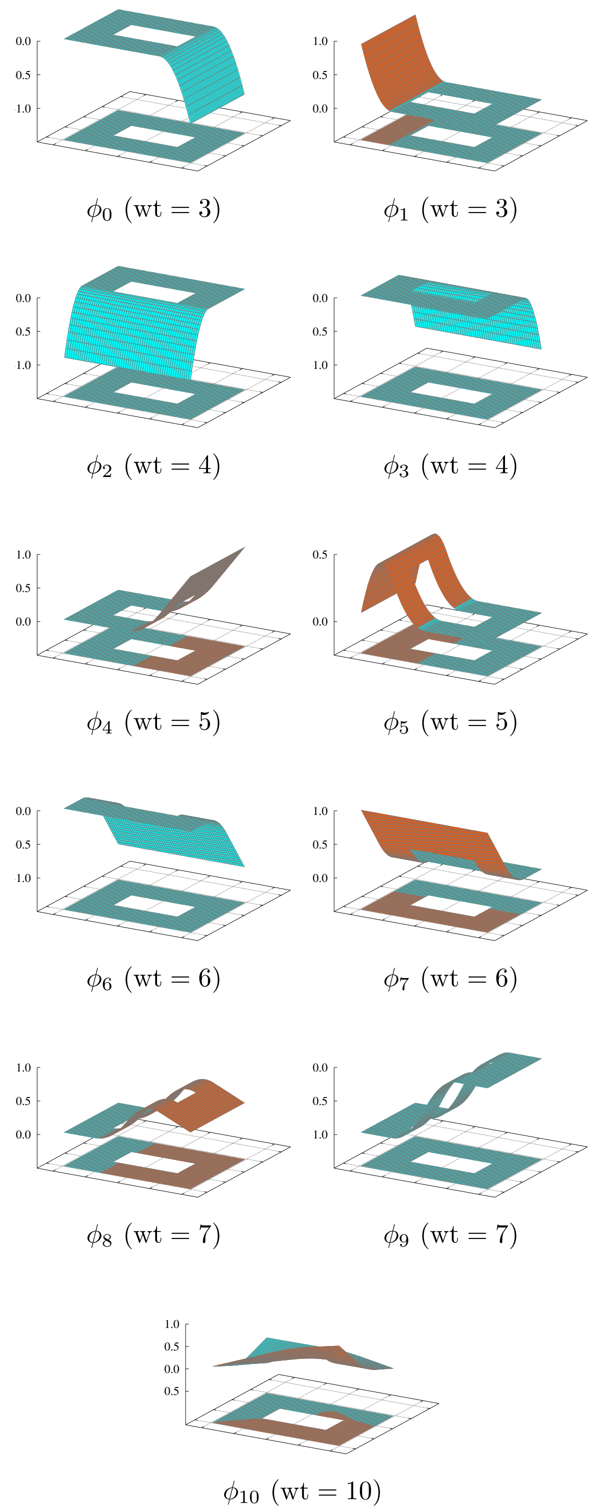


Figure 5: A minimum-weight basis ϕ for the space $\mathcal{P}_1^2[\mathcal{C}]$, where $\mathcal{C}$ is the mesh (2) in the figure 1.

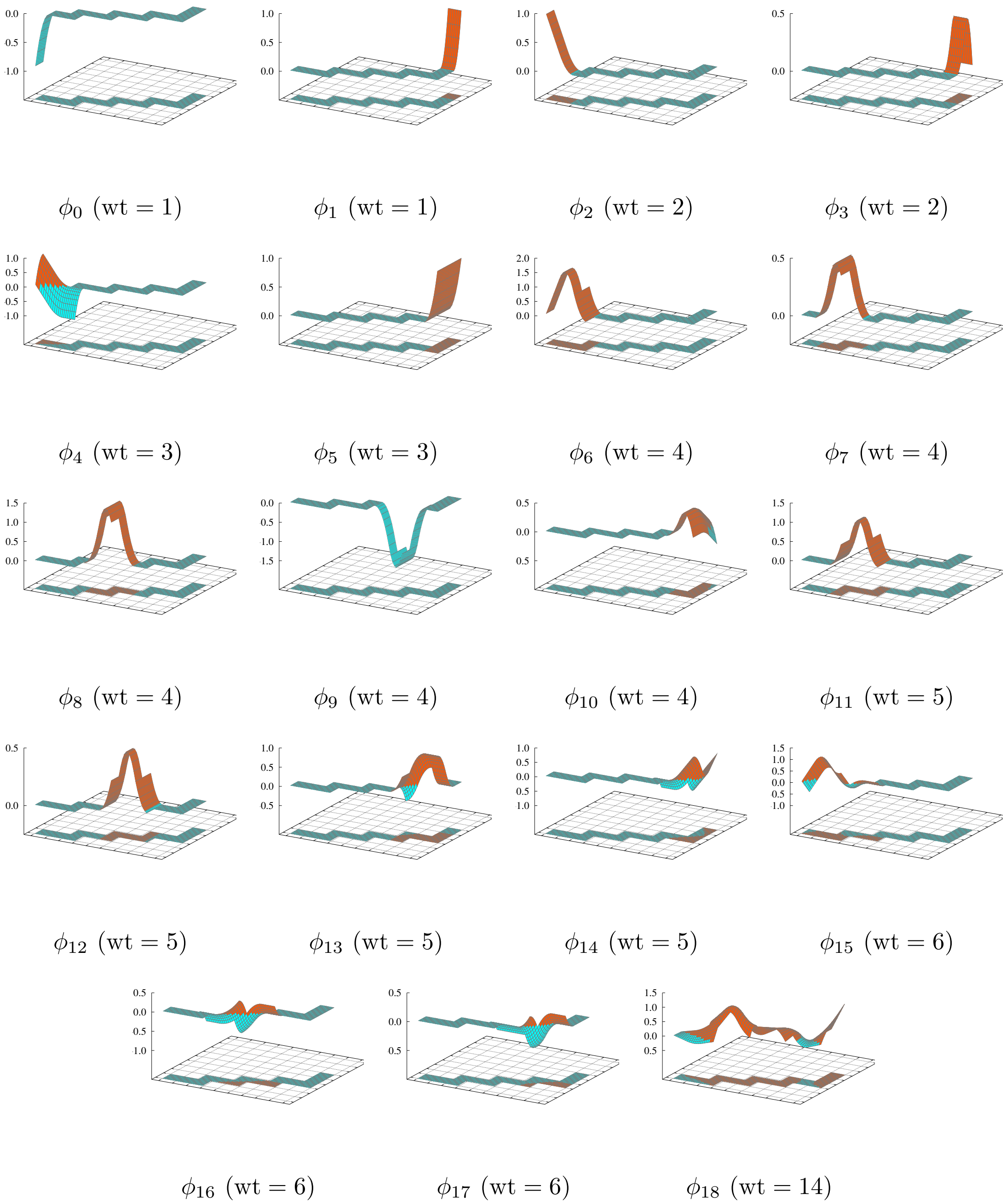


Figure 6: A minimum-weight basis ϕ for the space $\mathcal{P}_1^2[\mathcal{C}]$, where $\mathcal{C}$ is the mesh (3) of figure 1.