G1 Spline Functions for Point Cloud Fitting

Michelangelo Marsala, Angelos Mantzaflaris, Bernard Mourrain

G^1 Spline Functions for Point Cloud Fitting

Michelangelo Marsala*, Angelos Mantzaflaris, Bernard Mourrain

Inria Centre at Université Côte d'Azur, Sophia Antipolis, France

Abstract

In this work we present a new construction of basis functions that generate the space of geometrically smooth splines on an unstructured quadrilateral mesh. The basis is represented in terms of biquintic Bézier polynomials on each quadrilateral face. The gluing along the face boundaries is achieved using quadratic gluing data functions, leading to globally G^1–smooth spaces. We analyze the latter space and provide a combinatorial formula for its dimension as well as an explicit basis construction. Moreover, we assess the use of this basis in point cloud fitting problems. To apply G^1 least squares fitting, a quadrilateral structure as well as parameters in each quadrilateral is required. Even though the general problem of segmenting and parametrizing point clouds is beyond the focus of the present work, we describe a procedure that produces such a structure as well as patch-local parameters. Our experiments demonstrate the accuracy and smoothness of the obtained reconstructed models in several challenging instances.

Keywords: Point cloud fitting, multipatch domain, gluing data, geometrically smooth surfaces, spline basis

1. Introduction

A fundamental operation in geometric modeling is the construction of geometric models from measurements or observations. This consists in computing an accurate representation of the surface of an object, from a set of points obtained e.g. from laser scanners, Light Detection And Ranging (LIDAR) scanners or stereo camera devices. With the fast development of accessible acquisition tools, it is nowadays possible to use dense and precise point clouds to compute geometric models, such as meshes. Such reconstruction tools are indeed used in many applications including earth surveying tasks [51, 49, 44], geographic information and navigation systems [35], building model reconstructions [53], cultural heritage information systems [38], reverse engineering [10, 47], metrology [50, 54], medical imaging [42, 43].

Typically one adds structure to the point cloud by generating a mesh with the input points as vertices. Consequently, the mesh is further processed using mesh simplification, hole filing, noise removal, remeshing, smoothing, and so on, to obtain a representation of the shape with good regularity and good approximation properties (see e.g. [2, 23, 30, 5]). Nevertheless, a piecewise linear shape does not imply a smooth normal field therefore it is not always suitable for use in Computer Aided Geometric Design.

A classical approach for obtaining smooth shapes is based on B-spline or NURBS representations. Given a sequence of points a B-spline curve of prescribed regularity approximating the point set can be computed by solving a least square distance problem [11]. Iterating the procedure and inserting more knots results in an accurate, regular and efficient description of a curve. The approach can be extended to rectangular patches of tensor product B-splines for point clouds, which rely mainly on rectangular surface patches. Local refinement techniques, based on e.g. Truncated Hierarchical B-splines (THB) [13, 25], T-splines [24], Locally Refined splines (LR) [45], or Polynomial splines over Hierarchical T-meshes (PHT) [27] have also been developed to reduce the fitting error, while reducing the total number of required degrees of freedom.

The task of segmenting and parametrizing a point cloud is a necessary step towards fitting with spline functions that poses many challenges [52]. The choice of parameters impacts greatly the final fitting and several heuristics have been developed in that respect, based on local weights [12], optimization methods [37,

*Corresponding author

Email addresses: michelangelo.marsala@inria.fr (Michelangelo Marsala), angelos.mantzaflaris@inria.fr (Angelos Mantzaflaris), bernard.mourrain@inria.fr (Bernard Mourrain)

48], subdivision [3] and so on. We refer the reader to the recent review article [56] for a thorough presentation of different existing methods.

However, in many applications, the shape to be described cannot be mapped continuously and bijectively to a single rectangle and collections of quadrangular faces are needed to properly fit the whole point cloud. This does not prevent the reconstructed B-spline faces to have non-smooth junctions along shared edges. In [28], point-wise constraints on the normals of biquintic Bézier patches along the shared edges are imposed to obtain approximate regularity along the edges of the quadrangular patches. In [41] the authors propose a construction of G^1 biquintic B-splines starting from a given network of C^0 biquintic B-splines with single inner knots. This model construction can be seen as a projection to a G^1 model of the space of G^1 functions; however, it does not provide an explicit basis to that space that can be used for fitting problems. The paper [14] presents the construction of a G^1 biquintic Bézier surface interpolating the vertices of an input triangular mesh. Furthermore, newer biquintic construction as [22], provides the construction of G^2 surfaces developed via the use of functionals. Nevertheless, the above cited works only focus on the reconstruction of spline models and not on the computation of efficient bases for approximation purposes, which is the topic of the present work.

In [55], a so-called "immune genetic algorithm" is used to solve the fitting problem augmented with G^1 constraints. These patch-by-patch approaches require the treatment of G^1 conditions as additional constraints of the squared distance minimization problem and do not guarantee that the computed surface is globally smooth. In [15, 9], G^1 surfaces of bidegree $(3,3)$ and $(4,4)$ with 4 splits and respectively linear and quadratic gluing data are computed to interpolate a network of curves. Fitting optimization is used to approximate a point cloud, once the curve edges of the faces have been fixed. This constraints the resulting surface to have edges aligned with prescribed curves, and the fitting quality of the resulting surface depends on the chose of these curves.

The problem of computing dimension and explicit bases of spaces of G^1 functions is an active topic of research in spline theory and isogeometric analysis. Many authors have investigated the dimension and basis of spline spaces over planar domains. See e.g. [1, 17, 40, 26, 34] and references therein. Fewer results are available in the case of G^r, $r > 1$ spline spaces [33, 19, 20, 21, 7, 8].

Contributions. In this work, we describe a new construction of G^1 basis functions and show the applicability of these basis functions in fitting problems. The basis functions, computed by means of quadratic gluing data, are attached to vertices, edges and faces. Their coefficients in the Bernstein basis of each face depend solely on the topology of the mesh, therefore they can be computed once for every vertex valence. The fitting problem is solved using least square distance minimization in the space spanned by the G^1 spline basis associated with the quadrilateral mesh. The resulting surface is, by construction, smooth everywhere with continuous tangent planes along all shared edges. Our main focus is not on point cloud parametrization, which is, as already mentioned, a hard problem in itself but on the construction and analysis of good G^1 spline function spaces for approximating unstructured point clouds. Nevertheless, a computational approach is used to obtain parametrization (see Section 5.3): we triangulate the point cloud, then convert the fine triangulation into a coarse quadrilateral mesh, and finally we construct a G^1 spline surface supported on this coarse mesh using the approach in [32]. This initial surface is used to compute a footpoint for each point in the cloud and thereby associate parameters to be used in fitting.

Outline. In Section 2 we establish notation and present the definition of G^1 smooth spline on a multipatch domain making use of quadratic gluing data functions; moreover, we recall the equations generating the G^1 biquintic space that we are interested in. Section 3 is devoted to the construction of an explicit set of basis functions spanning the G^1 spline space, including explicit formulas for the computation of the Bézier representation of the basis functions. In Section 4 we prove that our construction yields a basis and we derive a combinatorial dimension formula for the space. We use the new bases in point cloud data fitting problems and elaborate on several examples in Section 5. In doing so, we detail on a computational approach to obtain a coarse quadrilateral structure and parameters for the point cloud (Section 5.3). Section 6 concludes the work.

2. G^1 functions on a mesh $\mathcal{M}$

Let us begin recalling some standard definitions. Given a matrix of points $\mathbf{b}_{i,j} \in \mathbb{R}^3$, we define a (tensor-product) Bézier patch as

$$\mathcal{Q}(u,v) = \sum_{i,j=0}^{n,m} \mathbf{b}_{i,j} B_{n,m}^{i,j}(u,v), \quad (u,v) \in [0,1]^2,$$

where $B^{i,j}_{n,m}(u,v) = B^i_n(u)B^j_m(v)$ is the bivariate Bernstein polynomials of bidegree (n,m), that is, $B^h_d(u) = \binom{d}{h}u^h(1-u)^{d-h}$.

A quadrilateral mesh in three dimensions is a collection of faces, edges and vertices together with their adjacency relations. Throughout this paper we work with meshes which contain only quadrilateral faces and we shall refer to such a structure as *quadrilateral mesh*, denoting it by $\mathcal{M}$.

An important notion is the valence of a vertex, which we define as the number of adjacent faces emanating from the vertex. An interior vertex is said to be regular (RV in short) in the case that the valence is equal to 4, and is called irregular (or extraordinary – EV) otherwise. In the particular case of a vertex on the mesh boundary, the regular case corresponds to a valence equal to 2. The case of valence 1, i.e. corresponding to a corner vertex, is treated separately. Given a mesh $\mathcal{M}$ it is interesting to count the main features it possesses: we will refer to n_V as the number of vertices of the mesh, which can be subdivided in n_{IEV} inner EVs, n_{BEV} boundary EVs and n_{RV} regular vertices. The total number of edges is denoted by n_E and is composed by the number of extraordinary and regular edges, n_{EE} and n_{RE} respectively, that is, edges attached to extraordinary and regular vertices, as well as the number of boundary edges n_{BE}. With n_F we refer to the number of faces and lastly n_C is the number of corners of the mesh. Moreover, a collection of patches $f = (f_\sigma)_{\sigma\in\mathcal{M}}$, with σ a face of the mesh $\mathcal{M}$ and $f_\sigma = f\big|_\sigma$ is called a multipatch surface [32].

A surface is said to be G^1 if it has continuous normals. In order to define the G^1 continuity between patches in a mathematical setting, we use *gluing data* functions $\mathfrak{a}_{N,N'} : e \longrightarrow \mathbb{R}$, $\mathfrak{b}_{N,N'} : e \longrightarrow \mathbb{R}$, with e the common edge between two adjacent patches of the mesh $\mathcal{M}$ linking a vertex of valence N with one of valence N'. In the following construction, we shall assume that each edge e is a curve on the surface parametrized in the interval $[0,1]$.

We say that a multipatch function $f = (f_\sigma)_{\sigma\in\mathcal{M}}$ on a mesh $\mathcal{M}$ is differentiable or G^1 if for any two functions f_0, f_1 defined on faces $\sigma_0, \sigma_1 \in \mathcal{M}$ sharing an edge e, we have (using a prescribed orientation of the coordinate frames as in [32])

$$\begin{cases} f_1(u_1,0) = f_0(0,u_1)\,, \\ \dfrac{\partial f_1}{\partial v_1}(u_1,0) = \mathfrak{b}_{N,N'}(u_1)\dfrac{\partial f_0}{\partial u_0}(0,u_1) + \mathfrak{a}_{N,N'}(u_1)\dfrac{\partial f_0}{\partial v_0}(0,u_1)\,, \end{cases} \qquad u_1 \in [0,1]\,, \tag{1}$$

where $\mathfrak{a}_{N,N'}, \mathfrak{b}_{N,N'}$ are the gluing data defined on the edge e. The functions satisfying eq. (1) for all shared edges belong to the linear space of G^1 functions on $\mathcal{M}$ with respect to the gluing data $\mathfrak{a}_{N,N'}$ and $\mathfrak{b}_{N,N'}$. In this paper we consider quadratic symmetric gluing data defined as

$$\begin{aligned} &\mathfrak{a}_{N,N'}(u) = a_0 B^0_2(u) - a_2 B^2_2(u)\,, \quad \text{where} \quad a_0 = 2\cos\left(\frac{2\pi}{N}\right)\,,\ a_2 = 2\cos\left(\frac{2\pi}{N'}\right)\,, \\ &\mathfrak{b}_{N,N'}(u) = -1\,, \end{aligned} \tag{2}$$

with $B^i_2(u) = \binom{2}{i}u^i(1-u)^{2-i}, i = 0,2$, the standard univariate Bernstein polynomials. We refer to [36, 4, 16, 9, 15, 32] for more details.

Here we assume that $N' = 4$ which implies $a_2 = 0$; geometrically speaking this means that the EVs of the mesh we consider are only linked to regular vertices. Let f_1 and f_2 be two Bézier patches of bidegree (d,d); the G^1 relations across an edge defined in (1) can be written in terms of the Bézier coefficients as

$$\begin{cases} \displaystyle\sum_{i=0}^{d} \mathrm{b}^{(1)}_{i,0} B^i_d(u) = \sum_{i=0}^{d} \mathrm{b}^{(0)}_{0,i} B^i_d(u)\,, \\ \displaystyle\sum_{i=0}^{d} \left(\mathrm{b}^{(1)}_{i,1} - \mathrm{b}^{(1)}_{i,0} + \mathrm{b}^{(0)}_{1,i} - \mathrm{b}^{(0)}_{0,i}\right) B^i_d(u) = a_0 B^0_2(u) \left(\sum_{i=0}^{d-1} \left(\mathrm{b}^{(0)}_{0,i+1} - \mathrm{b}^{(0)}_{0,i}\right) B^i_{d-1}(u)\right)\,, \end{cases} \tag{3}$$

where we used the gluing functions introduced in (2) and $\mathrm{b}^{(k)}_{i,j}$ are the control points of the patch σ_k. Throughout the paper we use biquintic Bézier patches, that is, $(d,d) = (5,5)$; this choice has also been made in [32], where the surface approximation scheme G^1ACC is presented. The G^1ACC surface is defined through masks returning a multipatch Bézier surface with G^1 smoothness around EVs and at least C^1 smoothness everywhere else. The coordinate functions of that parametrized surface are elements of the space of G^1 spline functions that we study in the present paper.

We are now in position to write the equations defining the geometric continuity constraints across an extraordinary edge, obtained from (3) for $d = 5$, as follows:

$$\begin{aligned}
\mathrm{b}_{0,1}^{(1)} + \mathrm{b}_{1,0}^{(0)} &= \bar{a}_0 \mathrm{b}_{0,0}^{(1)} + a_0 \mathrm{b}_{1,0}^{(1)}, && (4)\\
5(\mathrm{b}_{1,1}^{(1)} + \mathrm{b}_{1,1}^{(0)}) &= a_0 \mathrm{b}_{0,0}^{(1)} + 5\bar{a}_0 \mathrm{b}_{1,0}^{(1)} + 4a_0 \mathrm{b}_{2,0}^{(1)}, && (5)\\
10(\mathrm{b}_{2,1}^{(1)} + \mathrm{b}_{1,2}^{(0)}) &= -a_0 \mathrm{b}_{0,0}^{(1)} + 5a_0 \mathrm{b}_{1,0}^{(1)} + 10\bar{a}_0 \mathrm{b}_{2,0}^{(1)} + 6a_0 \mathrm{b}_{3,0}^{(1)}, && (6)\\
10(\mathrm{b}_{3,1}^{(1)} + \mathrm{b}_{1,3}^{(0)}) &= a_0 \mathrm{b}_{0,0}^{(1)} - 5a_0 \mathrm{b}_{1,0}^{(1)} + 10a_0 \mathrm{b}_{2,0}^{(1)} + 10\bar{a}_0 \mathrm{b}_{3,0}^{(1)} + 4a_0 \mathrm{b}_{4,0}^{(1)}, && (7)\\
\mathrm{b}_{4,1}^{(1)} + \mathrm{b}_{1,4}^{(0)} &= 2\mathrm{b}_{4,0}^{(1)}, && (8)\\
\mathrm{b}_{5,1}^{(1)} + \mathrm{b}_{1,5}^{(0)} &= 2\mathrm{b}_{5,0}^{(1)}, && (9)\\
10(\mathrm{b}_{3,0}^{(1)} - \mathrm{b}_{2,0}^{(1)}) &= \mathrm{b}_{0,0}^{(1)} - 5\mathrm{b}_{1,0}^{(1)} + 5\mathrm{b}_{4,0}^{(1)} - \mathrm{b}_{5,0}^{(1)}, && (10)
\end{aligned}$$

with $\bar{a}_0 = 2 - a_0$.

Starting from the above equations (4)-(10), we shall present an explicit construction of a set of basis functions generating the G^1 spline space and use it in point cloud fitting problems.

3. Bases extraction

In this section we present an explicit construction of the space of basis functions generating the G^1 space, which we refer to as $\mathcal{B}$. Following the topology of the input mesh $\mathcal{M}$, we can distinguish the space of functions attached to the vertices $\mathcal{B}_V$ (i.e. spanned by functions whose support lies on all the patches sharing the vertex), space of basis functions attached to the edges $\mathcal{B}_E$ (i.e. spanned by functions whose support lies on the patch(es) sharing the edge) and space of basis functions attached to the faces $\mathcal{B}_F$ (i.e. spanned by bases whose support lies only on the interior of a single patch). This decomposition will be exploited in the proofs of Section 4. Thus, we arrive at a set of basis functions which can be decomposed as

$$\mathcal{B} = \left(\bigcup_{i=1}^{n_V} \mathcal{B}_{V_i} \right) \cup \left(\bigcup_{i=1}^{n_E} \mathcal{B}_{E_i} \right) \cup \left(\bigcup_{i=1}^{n_F} \mathcal{B}_{F_i} \right). \tag{11}$$

We assume that all the EVs in $\mathcal{M}$ are isolated, that is, their one-ring neighborhood contains only RVs. Having EVs attached to regular vertices only, which translates into eq. (8)-(9), implies that vertex, edge and face functions have value and derivative equal to zero on the boundary of their support; thus basis functions with local support on a collection of faces of $\mathcal{M}$ do not influence functions in the region surrounding its support. By combining the equations in system (4)-(10) circularly around all the edges attached to an EV we can reformulate the G^1 constraints using a staircase block matrix, which is useful to better understand the bases extraction and their analysis which follows. Let $\mathbf{b}_{i,j} = \left(\mathrm{b}_{i,j}^{(k)} \right), k = 1, \ldots, N$, be the vector containing all the points $\mathrm{b}_{i,j}^{(k)}$ attached to the neighborhood of the EV we are considering, $\mathbf{u} = (\underbrace{1, \ldots, 1}_{N})^T$, $C \in \mathbb{R}^{N \times N}$ and the circulant matrix defined as

$$C = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \\ 1 & 0 & & \cdots & 0 \end{pmatrix} = \mathrm{Circ}(0, 1, 0, \ldots, 0). \tag{12}$$

Note that $\mathbf{b}_{4,0} = \mathbf{b}_{5,0} = 0$, as a consequence of the isolated EV hypothesis; the full system of G^1 relations around an EV can be written as

$$\begin{pmatrix} -\bar{a}_0\mathbf{u} & C_1 & & & & & & \\ -a_0\mathbf{u} & -5\bar{a}_0 I & -4a_0 I & 5C_2 & & & & \\ -\mathbf{u} & 5I & -10I & & 10I & & & \\ a_0\mathbf{u} & -5a_0 I & -10\bar{a}_0 I & & -6a_0 I & 10I & 10I & \\ -a_0\mathbf{u} & 5a_0 I & -10a_0 I & & 10\bar{a}_0 I & & 10I & 10I \end{pmatrix} \begin{pmatrix} \mathbf{b}_{0,0} \\ \mathbf{b}_{1,0} \\ \mathbf{b}_{2,0} \\ \mathbf{b}_{1,1} \\ \mathbf{b}_{3,0} \\ \mathbf{b}_{2,1} \\ \widehat{\mathbf{b}}_{1,2} \\ \mathbf{b}_{3,1} \\ \widehat{\mathbf{b}}_{1,3} \end{pmatrix} = \mathbf{0}\,, \tag{13}$$

where $a_0 = 2\cos(2\pi/N)$ is the first Bernstein coefficient of the gluing data function, $\bar{a}_0 = 2 - a_0$, $I \in \mathbb{R}^{N\times N}$ is the identity matrix,

$$C_1 = -a_0 I + C + C^{N-1}\,, \quad C_2 = I + C\,, \tag{14}$$

and $\widehat{\mathbf{b}}_{1,2} = C\,\mathbf{b}_{1,2}$, $\widehat{\mathbf{b}}_{1,3} = C\,\mathbf{b}_{1,3}$, with C in (12). The strategy we exploit to obtain the Bézier coefficients defining the basis functions is the following: starting from eq. (4)-(10) or eq. (13) we impose, one by one, the value one to each "free" coefficient involved in the G^1 constraints. Then, with this initial value we solve the G^1 relations in (4)-(10) or eq. (13), while we gradually set the value of any unconstrained coefficients that we encounter (i.e. coefficients which lead to an overdetermined equation) to zero.

3.1. The space $\mathcal{B}_V$: basis functions attached to a vertex

Here, we consider basis functions attached to inner and boundary vertices (extraordinary or regular) and corner vertices. We only provide the explicit construction for bases attached to inner extraordinary and inner regular vertices, since the construction for the remaining cases is analogous.

3.1.1. Construction of basis functions associated to an inner EV

Given an extraordinary vertex of valence N, we have attached to it basis functions concerning their value at the vertex and their value of the first and cross derivatives at the vertex. We will figure out during the construction that these bases are in number $N+3$.

Basis function attached to the vertex value. To extract the basis function associated to the value at the vertex, we start solving the system (4)-(10) with initial value $\mathbf{b}_{0,0} = 1$ and continuing the construction fixing zero values for all the control points which are not constrained by any relation we will encounter during the construction. With this assumption, we can rewrite eq. (4) in the form

$$C_1\mathbf{b}_{1,0} = (2 - a_0)\mathbf{u}\,, \tag{15}$$

with $C_1 \in \mathbb{R}^{N\times N}$ defined in (14). The solution of eq. (15) will return the values concerning the first derivatives of the basis with unit value at the vertex. As shown in [32], the matrix $C_1 = -a_0 I + C + C^{N-1}$ is singular and $\mathrm{corank}(C_1) = 2$; for this reason, in order to obtain a unique solution we need to insert two extra constraints to the system (15). Let $\mathrm{Ker}(C_1) = K = \mathrm{Span}\{\mathbf{k}_1, \mathbf{k}_2\}$ be the kernel of the matrix C_1 generated by the two vectors $\mathbf{k}_1$ and $\mathbf{k}_2$, we can choose our solution to be orthogonal to the space K, i.e. $\langle \mathbf{b}_{1,0}|\mathbf{k}_1\rangle = \langle \mathbf{b}_{1,0}|\mathbf{k}_2\rangle = 0$, where $\langle\cdot|\cdot\rangle$ represents the classical Euclidean scalar product. An explicit formula to compute the kernel K has been provided in [32]. This procedure allows us to achieve a unique solution for this set of control points. Going further in the resolution of the system, using the solution we just obtain Equation (4) and the circulant matrix C, and taking into account the constraint along the edge in eq. (10) which becomes

$$\mathbf{b}_{2,0} = \frac{1}{2}\mathbf{b}_{1,0} - \frac{1}{10}\mathbf{u}\,, \tag{16}$$

we can reorder eq. (5) as

$$C_2\mathbf{b}_{1,1} = -\frac{1}{5}a_0\mathbf{u} + 5(2 + a_0)\mathbf{b}_{1,0}\,, \tag{17}$$

with C_2 as in eq. (14). For odd values of the valence N the matrix C_2 is invertible, while for even occurrences we have $\mathrm{corank}(C_2) = 1$. To obtain a unique solution in the singular case we need to fix an extra constraint which we decide to be the orthogonality of the expected solution $\mathbf{b}_{1,1}$ to $\mathrm{Ker}(C_2)$. For a deeper explanation

of the solving strategy of (17) we refer the reader to [32]. Regarding the control points for the higher derivatives $\mathbf{b}_{2,1}$ and $\mathbf{b}_{3,1}$, from Equations (6)-(7) and using again (16) we come up with the relations

$$\mathbf{b}_{2,1} + \widehat{\mathbf{b}}_{1,2} = -\frac{1}{5}\mathbf{u} + \mathbf{b}_{1,0} \quad \text{and} \quad \mathbf{b}_{3,1} + \widehat{\mathbf{b}}_{1,3} = \mathbf{0} \tag{18}$$

which can be solved, for instance, imposing the extra relations $\mathbf{b}_{2,1} = \widehat{\mathbf{b}}_{1,2}$ and $\mathbf{b}_{3,1} = \widehat{\mathbf{b}}_{1,3}$. This procedure, as depicted from the construction, will return a unique basis function. In Fig. 1-(a) is presented an example of the coefficients obtained with the previous construction in case of an EV of valence $N = 3$.

Basis functions attached to the first derivatives at the vertex. To proceed with the construction of the second subset of basis functions we start again from Eq. (4) but imposing this time the value $\mathbf{b}_{0,0} = 0$; this choice leads to the following homogeneous linear system

$$C_1 \mathbf{b}_{1,0} = \mathbf{0}\,, \tag{19}$$

where the matrix C_1 is the same as the previous section and introduced in (14). A solution of eq. (19) is easily given by

$$\mathbf{b}_{1,0} \in \mathrm{Ker}(C_1) = \mathrm{Span}\{\mathbf{k}_1, \mathbf{k}_2\}$$

as for eq. (15); since the kernel of the matrix C_1 is a 2-dimensional space generated by the vectors $\mathbf{k}_1$ and $\mathbf{k}_2$ we will have two admissible solutions for the system (19) which, in fact, leads to two different bases function attached to value of the first derivative at the vertex obtained by solving the other G^1 relations starting with $\mathbf{b}_{1,0} = \mathbf{k}_1$ and $\mathbf{b}_{1,0} = \mathbf{k}_2$. The remaining constraints relating high order derivatives, taking into account the edge constraint obtained from eq. (10)

$$\mathbf{b}_{2,0} = \frac{1}{2}\mathbf{b}_{1,0}\,,$$

are given by the equations

$$\begin{aligned} C_2 \mathbf{b}_{1,1} &= \frac{1}{5}(2 + a_0)\mathbf{b}_{1,0}\,, \\ \mathbf{b}_{2,1} + \widehat{\mathbf{b}}_{1,2} &= \mathbf{b}_{1,0}\,, \\ \mathbf{b}_{3,1} + \widehat{\mathbf{b}}_{1,3} &= \mathbf{0}\,, \end{aligned}$$

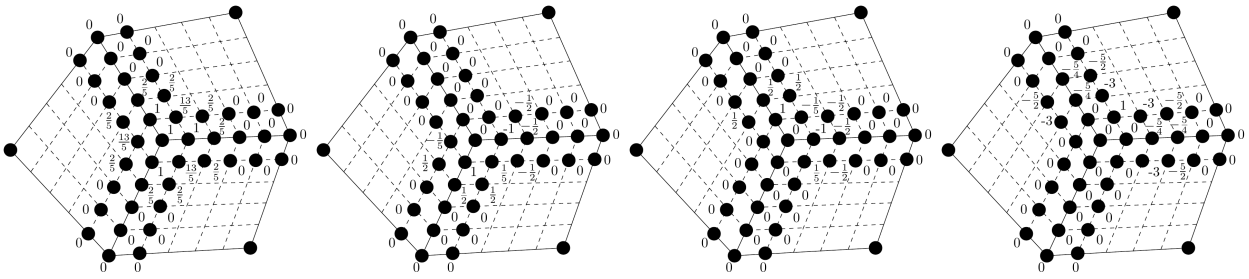
which can be solved in the same way as Eq. (17) and Eq. (18). Figure 1-(b) and (c) show the graph of the resulting functions for an EV with $N = 3$.

Basis attached to the cross derivatives at the vertex. To a vertex of valence N correspond N cross derivatives attached to it; this means that we need to compute N basis functions related to the value of the cross derivative at the vertex. Let's consider the k-patch belonging to the vertex ring. By setting the value $\mathrm{b}_{1,1}^{(k)} = 1$, from eq. (5) we realize that this point has only influence on values laying in patches $k-1$, k and $k+1$; more precisely, the points affected by this choice are only $\mathrm{b}_{2,0}^{(k)}$ and $\mathrm{b}_{2,0}^{(k+1)}$ regarding the second derivatives, which will have values equal to

$$\mathrm{b}_{2,0}^{(k)} = \mathrm{b}_{2,0}^{(k+1)} = \frac{5}{4a_0}\,,$$

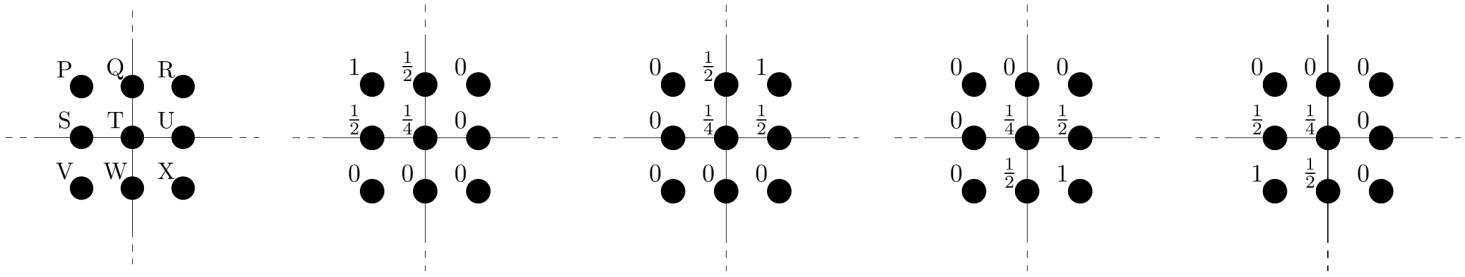
being well defined since $a_0 \neq 0 \iff N \neq 4$, that is the case of an EV we are in fact investigating. Regarding higher order derivatives, we have the points $\mathrm{b}_{3,0}^{(k)}, \mathrm{b}_{3,0}^{(k+1)}, \mathrm{b}_{1,2}^{(k-1)}, \mathrm{b}_{2,1}^{(k)}, \mathrm{b}_{1,2}^{(k)}, \mathrm{b}_{2,1}^{(k+1)}$ and $\mathrm{b}_{1,3}^{(k-1)}, \mathrm{b}_{3,1}^{(k)}, \mathrm{b}_{1,3}^{(k)}, \mathrm{b}_{3,1}^{(k+1)}$ defined by the relations

$$\begin{aligned} \mathrm{b}_{3,0}^{(k)} = \mathrm{b}_{3,0}^{(k+1)} &= \frac{5}{4a_0}\,, \\ \mathrm{b}_{1,2}^{(k-1)} + \mathrm{b}_{2,1}^{(k)} = \mathrm{b}_{2,1}^{(k)} + \mathrm{b}_{1,2}^{(k+1)} &= \frac{1}{2}\left(\frac{5}{a_0} - 1\right)\,, \\ \mathrm{b}_{1,3}^{(k-1)} + \mathrm{b}_{3,1}^{(k)} = \mathrm{b}_{3,1}^{(k)} + \mathrm{b}_{1,3}^{(k+1)} &= \frac{5}{2a_0}\,, \end{aligned}$$



(a) (b) (c) (d)

Figure 1: Coefficients for a basis function attached to the value at the vertex (a), attached to the first derivatives at the vertex (b)-(c) and attached to the cross derivatives at the vertex (d) for an EV of valence $N = 3$.



(a) (b)

Figure 2: (a): labelling of the control points around an inner regular vertex. (b): values for the Bézier points of the four basis functions.

which are obtained making use of Eq. (6)-(7) and (10). Similarly to the other situations, the previous equations can be solved as in (18). Repeating the same construction for all the patches in the ring we come up with the N basis functions attached to the cross derivatives. The result in the case $N = 3$ is shown in Fig.1-(d), while Fig. 4 presents the set of basis functions for an EV of valence $N = 5$.

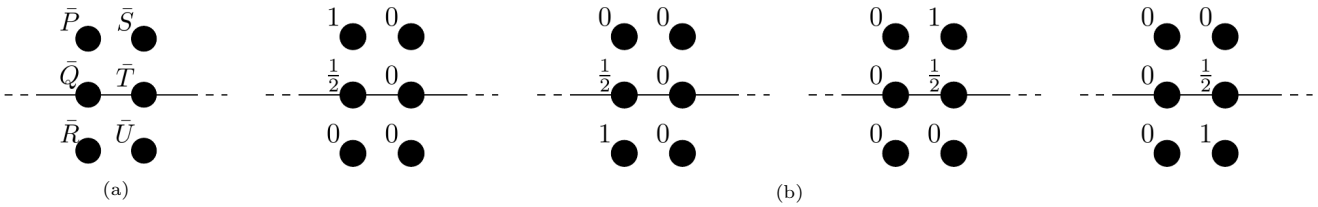
3.1.2. Basis functions at an inner regular vertex

In presence of a regular vertex (RV), i.e. a vertex with valence $N = 4$, by construction we expect to have C^1 regularity for these bases; this is in fact what eq. (8)-(9) state.

First we need to investigate how many RV bases we have. To do that we first need to expand cyclically eq. (8)-(9) to all the control points around the vertex; this procedure, using the notation in Fig. 2-(a) leads to the following system:

$$\begin{pmatrix} 1 & 0 & 0 & -2 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -2 & 0 & 0 & 1 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -2 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 \end{pmatrix} \begin{pmatrix} \mathrm{P} \\ \mathrm{Q} \\ \mathrm{R} \\ \mathrm{S} \\ \mathrm{T} \\ \mathrm{U} \\ \mathrm{V} \\ \mathrm{W} \\ \mathrm{X} \end{pmatrix} = \mathbf{0}\,. \tag{20}$$

Let D be the coefficient matrix in (20). The number of bases attached to the RV is given by $\mathrm{corank}(D) = 9 - \mathrm{rank}(D) = 9 - 5 = 4$. Hence, a choice for the Bézier points returning linearly independent bases (by construction) verifying eq. (20) is given by the coefficients in Fig. 2-(b). In presence of RV connected to an EV is also needed to modify more control points for each extraordinary edge, according to eq. (6) and (7).



(a) (b)

Figure 3: (a): labelling of the control points across a regular inner edge. (b): values for the Bézier points of the four basis functions.

3.1.3. Bases linked to extraordinary and regular boundary vertices and corners

The extraction of these types of bases is analogous to the constructions developed in the previous sections for the inner cases. Naming with κ the valence of an extraordinary boundary vertex, i.e. the number of patches attached to it, imitating the process in Section 3.1.1 we come up with $\kappa+3$ bases functions, which is equivalent to $N+2$ since $\kappa = N-1$. On the other hand, to obtain the bases bounded to a regular boundary vertex and corner we need to copy the procedure shown in Section 3.1.2: in both cases, following the same strategy leading to eq. (20), we come to get to obtain also in this case four bases functions.

3.2. Basis functions $\mathcal{B}_E$ attached to an edge

In this second set of functions, we have bases attached to inner and boundary edges, either extraordinary and regular ones. In the same way as the vertex bases we will present the explicit construction in the case of inner extraordinary and regular edge functions, whereas the construction for the remaining case comes straightforwardly.

3.2.1. Construction of bases connected to an extraordinary edge

The forming of these functions is obtained starting from Equations (6)-(7) and (10). Similarly to the construction in Section 3.1.1, to extract the Bézier coefficients for the bases, we need to set zero values at the free points appearing in the equations; in the construction of bases connected to extraordinary edges, the control points we need to nullify are all the points laying on the edge, i.e. $\mathbf{b}_{0,0}$, $\mathbf{b}_{1,0}$, $\mathbf{b}_{2,0}$, $\mathbf{b}_{3,0}$, $\mathbf{b}_{4,0}$ and $\mathbf{b}_{5,0}$. This assumption transforms Eq. (6)-(7) into

$$\begin{aligned} \mathbf{b}_{2,1} + \widehat{\mathbf{b}}_{1,2} &= \mathbf{0}\,, \\ \mathbf{b}_{3,1} + \widehat{\mathbf{b}}_{1,3} &= \mathbf{0}\,, \end{aligned} \tag{21}$$

which define the two basis functions living on an extraordinary edge. The simplest solution satisfying eq. (21) is to take $\mathbf{b}_{2,1} = \mathbf{b}_{3,1} = 1$ and $\widehat{\mathbf{b}}_{1,2} = \widehat{\mathbf{b}}_{1,3} = -1$, or vice-versa.

3.2.2. Basis functions belonging to an inner regular edge

These basis functions are obtained with a similar approach as the bases in Section 3.1.2 for an inner regular vertex; the points involved in this construction are the two pairs of layers in the inner part of the edge, i.e. away from the influence of the vertices' equations. In order to determine the number of these bases, applying the C^1 constraints in Eq. (8) and Eq. (9) to the two layers of control points implicated in this analysis with the notation given by Fig. 3-(a) we have:

$$\begin{pmatrix} 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 \end{pmatrix} \begin{pmatrix} \bar{\mathrm{P}} \\ \bar{\mathrm{Q}} \\ \bar{\mathrm{R}} \\ \bar{\mathrm{S}} \\ \bar{\mathrm{T}} \\ \bar{\mathrm{U}} \end{pmatrix} = \mathbf{0}\,. \tag{22}$$

If $\bar{D}$ is the matrix in (22), the number of basis functions attached to an inner regular edge is given by $\mathrm{corank}(\bar{D}) = 4$ and a set of possible solutions verifying these constraints returning linearly independent bases is given by the configurations in Fig. 3-(b).

3.2.3. Boundary edge basis functions

Since we are in presence of a boundary edge, in this situation we have no smoothness constraints to impose. Hence, the basis functions in this case are the classical bivariate Bézier polynomials obtained assigning the unit value to the four control points involved in this setting, one at the time, to obtain the four bases we were looking for.

3.3. Basis functions $\mathcal{B}_F$ attached to a face

To conclude the construction of our space of bases we miss to the define the basis functions belonging uniquely to a single patch. As for the case of boundary edge functions, here we have no regularity conditions to impose; thus the construction is the same as in Section. 3.2.3 returning four Bézier polynomial on each face.

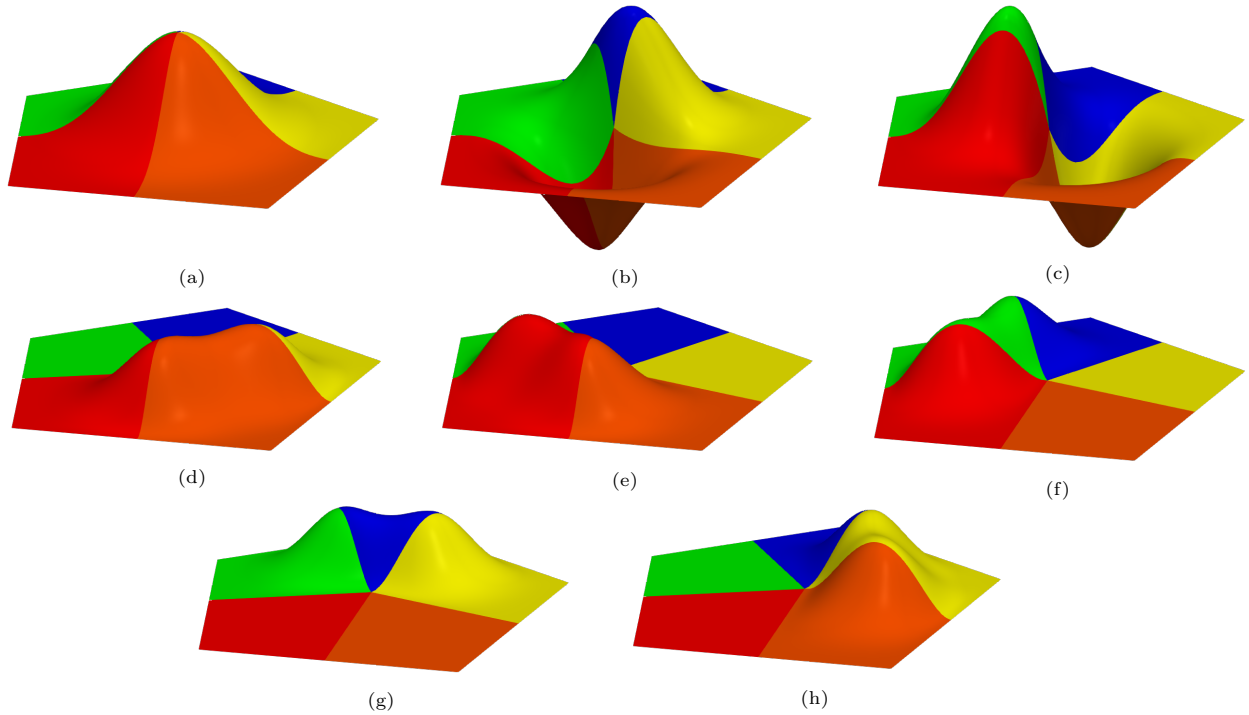


Figure 4: Basis function attached to the value of the vertex (a), value of first derivatives (b)-(c) and value of cross derivatives (d)-(e)-(f)-(g)-(h) for an EV of valence $N = 5$.

4. Analysis of the bases and space dimension

The functions we built in Section 3 form a basis of the spline space $\mathbb{G}^1(\mathcal{M})$ of G^1 functions over the mesh $\mathcal{M}$. In this section a proof for this statement is provided as well as a dimension formula for the same space. The first part of the proof, led by contradiction, focuses on the linear independence of the basis functions and it is carried out making advantage of the property (11) and analyzing their supports; the second half, instead, uses the equations defining the smoothness constraints of the basis functions in order to prove their generating property.

Theorem 1. *The functions constructed in Section 3 form a set of bases $\mathcal{B}$ for the space $\mathbb{G}^1(\mathcal{M})$ over a quad mesh $\mathcal{M}$.*

Proof. In order to prove that the set $\mathcal{B}$ is a basis of our space we need to prove their linear independence and the property to be a generating set. For the linear independence, we proceed by contradiction. Since we assume to deal with isolated EVs only, we have by construction that the supports of vertex basis functions

attached to different vertices are disjoint; the same property holds for edge basis functions defined over different edges as well as face basis functions attached to different faces. Hence, we can restrict our proof on basis functions attached on elements sharing a common vertex. Therefore we consider an EV of valence N with its N extraordinary edges and N faces (the case with regular vertices and regular edges is analogous) and we assume that there exists a linear dependence between the basis functions supported on these elements, i.e. there exist non zero coefficients $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}$ such that

$$\sum_{i=1}^{N+3} \alpha_i B_i^{EV} + \sum_{j=1}^{N} \sum_{i=1}^{2} \beta_i^{EE_j} B_i^{EE_j} + \sum_{j=1}^{N} \sum_{i=1}^{4} \gamma_i^{F_j} B_i^{F_j} = 0\,. \tag{23}$$

From (23) it holds that

$$\operatorname{supp}\left(\sum_{i=1}^{N+3} \alpha_i B_i^{EV} + \sum_{j=1}^{N} \sum_{i=1}^{2} \beta_i^{EE_j} B_i^{EE_j}\right) = \operatorname{supp}\left(-\sum_{j=1}^{N} \sum_{i=1}^{4} \gamma_i^{F_j} B_i F_j\right),$$

and in particular

$$\operatorname{supp}\left(\sum_{i=1}^{N+3} \alpha_i B_i^{EV} + \sum_{j=1}^{N} \sum_{i=1}^{2} \beta_i^{EE_j} B_i^{EE_j}\right) \subseteq \left(\bigcup_{i=1}^{N+3} \operatorname{supp}\left(B_i^{EV}\right)\right) \cup \left(\bigcup_{i=1}^{N}\left(\bigcup_{j=1}^{2} \operatorname{supp}\left(B_i^{EE_j}\right)\right)\right)$$

$$\operatorname{supp}\left(\sum_{i=1}^{N+3} \alpha_i B_i^{EV} + \sum_{j=1}^{N} \sum_{i=1}^{2} \beta_i^{EE_j} B_i^{EE_j}\right) \subseteq \left(\bigcup_{i=1}^{N}\left(\bigcup_{j=1}^{4} \operatorname{supp}\left(B_i^{F_j}\right)\right)\right).$$

It follows that

$$\operatorname{supp}\left(\sum_{i=1}^{N+3} \alpha_i B_i^{EV} + \sum_{j=1}^{N} \sum_{i=1}^{2} \beta_i^{EE_j} B_i^{EE_j}\right)$$

$$\subseteq \left(\left(\bigcup_{i=1}^{N+3} \operatorname{supp}\left(B_i^{EV}\right)\right) \cup \left(\bigcup_{i=1}^{N}\left(\bigcup_{j=1}^{2} \operatorname{supp}\left(B_i^{EE_j}\right)\right)\right)\right) \cap \left(\bigcup_{i=1}^{N}\left(\bigcup_{j=1}^{4} \operatorname{supp}\left(B_i^{F_j}\right)\right)\right) = \varnothing\,. \tag{24}$$

Hence, (24) implies

$$\sum_{i=1}^{N+3} \alpha_i B_i^{EV} + \sum_{j=1}^{N} \sum_{i=1}^{2} \beta_i^{EE_j} B_i^{EE_j} = 0 = \sum_{j=1}^{N} \sum_{i=1}^{4} \gamma_i^{F_j} B_i^{F_j}\,,$$

that is $\gamma_i^{F_j} = 0\ \forall\, i, j$ since $B_i^{F_j}$ are linearly independent by construction. Now we have

$$\sum_{i=1}^{N+3} \alpha_i B_i^{EV} = -\sum_{j=1}^{N} \sum_{i=1}^{2} \beta_i^{EE_j} B_i^{EE_j}\,. \tag{25}$$

Let us fix two adjacent patches k and $k-1$ sharing the EV we are considering and let us focus on the control points $\mathrm{b}_{0,0}^{(k)}, \mathrm{b}_{1,0}^{(k)}, \mathrm{b}_{0,1}^{(k)}, \mathrm{b}_{1,1}^{(k)}, \mathrm{b}_{1,0}^{(k-1)}, \mathrm{b}_{1,1}^{(k-1)}$. By definition, these control points are always equal to zero for all the basis functions $B_i^{EE_j}$, while this is not the case for the B_i^{EV}; hence, if we now look at the submatrices corresponding to the points we are considering for both B_i^{EV} and $B_i^{EE_j}$, we deduce that in the first case the selected submatrix is invertible, while the second submatrix is the null matrix. Therefore it follows that $\forall i$, $\alpha_i = 0$ since the B_i^{EV} are linearly independent by construction and, consequently, using the same argument, $\beta_i^{EE_j} = 0\ \forall i, j$.

Regarding the generating property, let $f \in \mathbb{G}^1(\mathcal{M})$. Since the basis functions $B_i^{F_j}$ attached to a face F_j are C^1 smooth we can define the function

$$f' := f - \sum_{j=1}^{n_F} \sum_{i=1}^{4} c_i^{F_j} B_i^{F_j}\,, \tag{26}$$

which is still G^1 and is such that the inner face coefficients vanish. The same procedure can be applied to the C^1 basis function attached to corners $B_i^{C_j}$, boundary edges $B_i^{BE_j}$ and regular edges $B_i^{RE_j}$ which can be used to define a new function starting from (26) as

$$f^{''} := f^{'} - \sum_{j=1}^{n_C}\sum_{i=1}^{4} c_i^{C_j} B_i^{C_j} - \sum_{j=1}^{n_{BE}}\sum_{i=1}^{4} c_i^{BE_j} B_i^{BE_j} - \sum_{j=1}^{n_{RE}}\sum_{i=1}^{4} c_i^{RE_j} B_i^{RE_j} \in G^1\,, \tag{27}$$

so that it has null coefficients in the above mentioned sectors.

Let $\mathbf{s}$ be the indices of the 9 control points in Figure 2 related to the bases $B_i^{RV_j}$ attached to a regular vertex RV_j and let denote by $f[\mathbf{s}]$ the corresponding control coefficients of the function f. Since from (20) $A\,f[s] = 0$, it follows that $\mathrm{Ker}(A) = \mathrm{Span}\left\{B_i^{RV_j}[\mathbf{s}]\right\}$. Hence $f^{''}[\mathbf{s}] = \sum_{i=1}^{4} c_i^{RV_j} B_i^{RV_j}[\mathbf{s}]$. We proceed similarly with boundary EVs $B_i^{BEV_j}$ and boundary regular vertices $B_i^{BRV_j}$ to define respectively coefficients $c_i^{BEV_j}, c_i^{BRV_j}$ so that the function

$$f^{'''} := f^{''} - \sum_{j=1}^{n_{RV}}\sum_{i=1}^{4} c_i^{RV_j} B_i^{RV_j} - \sum_{j=1}^{n_{BRV}}\sum_{i=1}^{4} c_i^{BRV_j} B_i^{BRV_j} - \sum_{j=1}^{n_{BEV}}\sum_{i=1}^{N_{BEV_j}+2} c_i^{BEV_j} B_i^{BEV_j} \tag{28}$$

is still a G^1 smooth function and it has zero coefficients in the previously treated regions.

Now, the function $f^{'''}$ has only non-zero coefficients around extraordinary vertices and their incident edges. We define $\mathbf{t}$ as the indices of the non-zero coefficients for an extraordinary vertex EV_j (black points in Fig. 1). Similarly to the case of regular vertices, from Eq. (13), $\bar{A}\,f^{'''}[\mathbf{t}] = 0$, where $\bar{A}$ is the matrix in (13) containing the G^1 constraints. Then, $\mathrm{Ker}(\bar{A}) = \mathrm{Span}\left\{B_i^{EV_j}[\mathbf{t}]\right\}$ which leads to $f^{'''}[\mathbf{t}] = \sum_{i=1}^{3N_{EV_j}+3} c_i^{EV_j} B_i^{EV_j}[\mathbf{t}]$. Finally, we have that

$$f^{''''} := f^{'''} - \sum_{j=1}^{n_{EV}}\sum_{i=1}^{3N_{EV_j}+3} c_i^{EV_j} B_i^{EV_j} = 0\,,$$

which concludes the proof.

□

As consequence of the structure of the bases set $\mathcal{B}$ we have the following formula for the dimension of our spline space.

Corollary 1. *The space $\mathbb{G}^1(\mathcal{M})$ has dimension given by:*

$$\begin{aligned}\dim\left(\mathbb{G}^1(\mathcal{M})\right) &= \sum_{i=1}^{n_V}|\mathcal{B}_{V_i}| + \sum_{i=1}^{n_E}|\mathcal{B}_{E_i}| + \sum_{i=1}^{n_F}|\mathcal{B}_{F_i}| \\ &= \sum_{i=1}^{n_{EV}} N_{EV_i} + 3n_{IEV} + 2(n_{BEV}+n_{EE}) + 4(n_{RE}+n_{BE}+n_{RV}+n_C+n_F)\,.\end{aligned}$$

Proof. The proof is obtained using the decomposition in (11) and summing up all the basis functions for each feature of the mesh shown in Sections 3.1, 3.2 and 3.3. □

5. Numerical experiments

We present numerical experiments in which we assess the basis functions constructed in Section 3 in point cloud fitting problems. The setup of our investigation is the classical least square fitting problem: given a point cloud $\mathcal{P}$, i.e. a set of points $P_i \in \mathbb{R}^3$, $i = 1, \ldots, n_P$, with associated parameters $(u_i, v_i) \in [0,1]^2$ on the patch ℓ_i, we want to find the coefficients $c_i \in \mathbb{R}^3$ of a G^1 surface $\boldsymbol{G} = \sum_k c_k B_k$ such that the quantity

$$F = \sum_{i=1}^{n_P} \|\boldsymbol{G}^{(\ell_i)}(u_i, v_i) - P_i\|_2^2 + \lambda E_{thin}\,, \quad \lambda \geq 0\,, \tag{29}$$

is minimal, where $\boldsymbol{G}^{(\ell)}$ $(\ell = 1, \ldots, n_F)$ is the geometry map introduced in Sec. 2

$$\boldsymbol{G}^{(\ell)} : [0,1]^2 \longrightarrow \mathbb{R}^3\,, \quad (u_i^{(\ell)}, v_i^{(\ell)}) \longmapsto \boldsymbol{G}^{(\ell)}(u_i^{(\ell)}, v_i^{(\ell)}) = \sum_{i,j=1}^{36} \mathrm{b}_{i,j}^{(\ell)} B_{i,j}^{(\ell)}(u_i^{(\ell)}, v_i^{(\ell)})\,, \tag{30}$$

with $\mathrm{b}_{i,j}^{(\ell)}$ control points in the standard bivariate Bernstein polynomials $B_{i,j}^{(\ell)}$ defining the ℓ-th patch $\Omega^{(\ell)} = \boldsymbol{G}^{(\ell)}([0,1]^2)$.

We also take into account in our minimization problem (29) an energy term given by the standard thin-plate energy

$$E_{thin} = \sum_{\ell=1}^{n_F} \iint_{[0,1]^2} \left\|\boldsymbol{G}_{uu}^{(\ell)}\right\|_2^2 + 2\left\|\boldsymbol{G}_{uv}^{(\ell)}\right\|_2^2 + \left\|\boldsymbol{G}_{vv}^{(\ell)}\right\|_2^2 du\, dv\,, \tag{31}$$

which is controlled by the parameter λ. The minimization of the functional in (31) is responsible for a regularization effect on the final surface $\boldsymbol{G}$ (i.e. oscillations). Moreover, following the construction presented in [18], we perform few iterations of parameter correction to further reduce the approximation error, if needed.

Since our focus is on the quality of the G^1 basis for fitting, the parametrized data that we use in the following numerical experiments are obtained by evaluating certain input functions or surfaces to obtain parameters. In particular, starting from a planar mesh, in order to build point cloud data from a given analytic function f, we compute parameters (u_i, v_i) by randomly sampling a certain number of points in the unit square $[0,1]^2$, for each patch. Therefore, the point cloud is obtained by the triple $(x_i, y_i, f(x_i, y_i))$, with $(x_i, y_i) = \boldsymbol{G}^{(\ell_i)}(u_i, v_i)$. A similar procedure is performed to obtain a point cloud from a quad mesh yielding an ACC surface [29]: considering the map $\boldsymbol{G}_{ACC}$, we obtain sample points $(x_i, y_i, z_i) = \boldsymbol{G}_{ACC}^{(\ell_i)}(u_i, v_i)$ randomly on the surface, with $(u_i, v_i) \in [0,1]^2$. Furthermore, we present two examples of data fitting starting from point clouds without any given structure.

After having computed a least square surface $\boldsymbol{G}$ from a point cloud $\mathcal{P}$ let define the array of errors $\mathbf{err} = \{err_i\}_{i=1}^{nP}$ whose entries are the quantities $err_i = \|\boldsymbol{G}^{(\ell_i)}(u_i, v_i) - P_i\|_2$, i.e. the Euclidean distance (ℓ_2 norm) from each point of the cloud and the corresponding value on the surface evaluated in its parameter. From this we define two quantities which will be used to assess the accuracy of the fitting, namely the biggest error and the root mean squared error (RMSE in short):

$$L^\infty := \max_{i=1,\ldots,n_P} err_i\,, \quad \text{RMSE} := \sqrt{\frac{1}{n_P}\sum_{i=1}^{n_P} err_i^2}\,. \tag{32}$$

5.1. Point cloud by analytic function evaluation

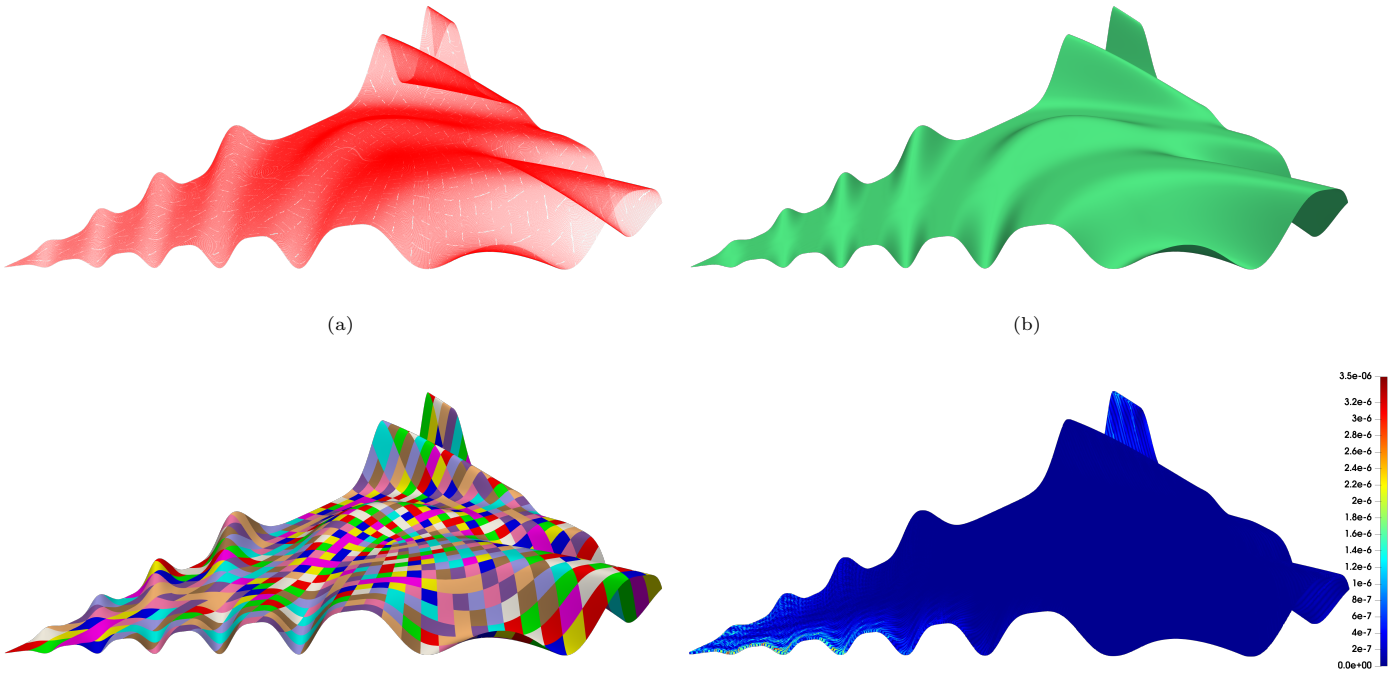
We present here experiments which are developed using point cloud data obtained by sampling an input function $f(x, y)$ over a domain identified by a quad mesh $\mathcal{M}$. In Example 1 we focus the attention on the goodness of the fitting showing how it improves when the number of basis functions increases, without the need to use extra smoothness constraints. On the other hand, Example 2 exhibit the power of our construction when thin plate energy is used in order to obtain an optimal result, but without increasing the dimension of the spline space.

Example 1. For this test, the point cloud is obtained evaluating the function

$$f_T(x, y) = \frac{y}{2\left(\cos(4(x^2 + y - 1))\right)}\,, \quad (x, y) \in \mathcal{M}_T \tag{33}$$

over the mesh $\mathcal{M}_T$, defining a triangle, formed by 3 patches which presents an EV of valence $N = 3$; our sampling produced a point cloud formed by 1536000 points. This example uses no smoothing property,

which means we fix $\lambda = 0$. From the result in Figure 5-(a) can be noticed that constructing our spline space over a coarse mesh as $\mathcal{M}_T$ leads to a fitting surface which is not approximating in a proper way our point set, due to its very oscillating behavior. This issue can be solved by increasing the number of bases generating the spline space, i.e. increasing the number of patches defining the polygonal domain; in this example the mesh has been refined via Catmull-Clark subdivision. In Tab. 1 and Fig. 6 are respectively listed and plotted the errors computed from (32) for 5 refinement levels. Fig. 5 shows the input point cloud (a) and the results at the last step of subdivision (b)-(c) together with its error color plot (d), while Tab. 2 and Fig. 6 presents the resulting errors, when the number of points increase with the dimension of the G^1 spline space. As shown in Fig. 6, we recovered the optimal rate of convergence 6 using biquintic splines.



(a) (b)

(c) (d)

Figure 5: (a): final point cloud obtained sampling (33) at the last refinement step. (b): resulting surface. (c): multipatch representation of the surface. (d): error color plot of the ℓ_2 distances. Bounding box with maximum length: 2.

$\dim\left(\mathbb{G}^1(\mathcal{M}_T)\right)$	72	240	864	3264	12672
L^∞ **error**	0.364e-00	0.122e-00	0.834e-01	0.178e-01	0.790e-02
RMSE	0.491e-01	0.149e-01	0.335e-02	0.353e-03	0.570e-04

Table 1: Maximal error L^∞ and RMSE for the surfaces in Example 1 obtained under 5 Catmull-Clark subdivision steps from a point cloud of 150528 elements.

$\dim\left(\mathbb{G}^1(\mathcal{M}_T)\right)$	72	240	864	3264	12672	49920
L^∞ **error**	0.225e-00	0.844e-01	0.340e-01	0.385e-02	0.187e-03	0.351e-05
RMSE	0.349e-01	0.140e-01	0.338e-02	0.256e-03	0.723e-05	0.119e-06

Table 2: Maximal error L^∞ and RMSE for the surfaces in Example 1 obtained under 6 Catmull-Clark subdivision steps with dyadically increased numbers of elements in the point cloud to estimate the rate of convergence. The point clouds have respectively 1500, 6000, 24000, 96000, 384000 and 1536000 points.

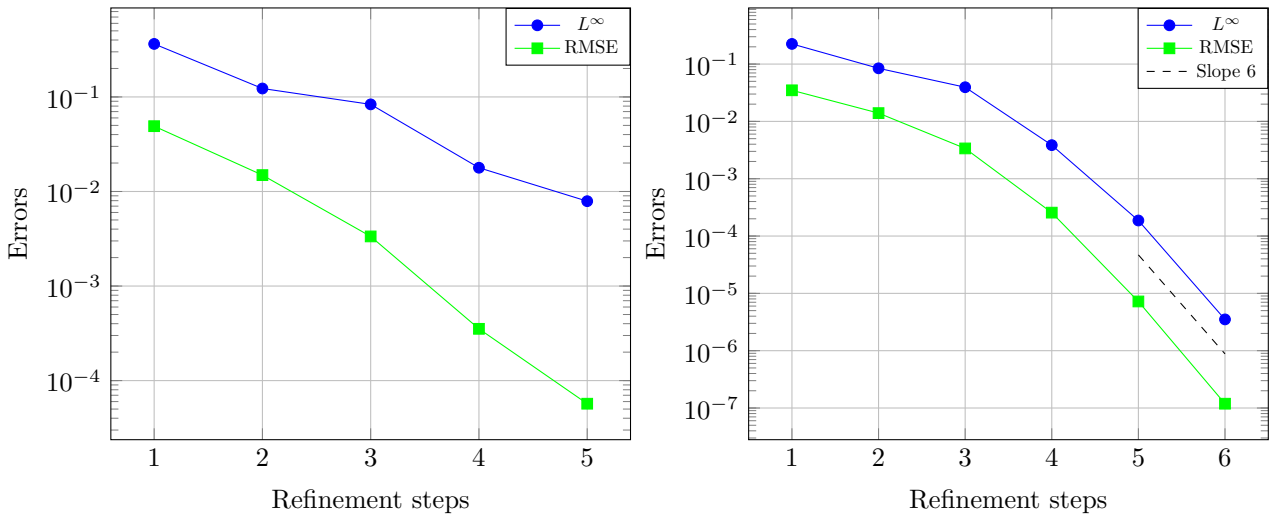


Figure 6: Left: experimental behavior of the errors (Table 1). Right: estimated rate of convergence (Table 2).

Example 2. Here, the cloud data is derived sampling the function

$$f_E(x,y) = \sum_{\nu\in\{-3,0,3\}} \frac{2}{3e^{\sqrt{(10x+\nu)^2+(10y+\nu)^2}}}, \quad (x,y)\in\mathcal{M}_E\,, \tag{34}$$

where $\mathcal{M}_E$ is a quad mesh tracing out an hexagon composed of 96 patches identifying an EV of valence $N=6$ in its middle. The point cloud we obtain is formed of 153600 points. Here we show the power of the smoothing property of this construction: differently from Example 1, fixing the number of patches i.e. the number of basis function, which is equal to 1725 in our case, we compute the fitting surface increasing the smoothing parameter λ from 10^{-3} to 10^{-1} with a step of 10^{-1}. From Fig. 7-(b) we notice that the fitting surface presents several wrinkles around the middle peak; by increasing the smoothing factor λ we recover regularity in the output function which presents much less irregularities, as can be noticed in Fig. 7-(d). As expected, this procedure will produce at every iteration a smoother function than the previous, but on the other side this forced regularity constraint is reflected in an increase of the three errors in (32); this phenomenon is shown in Tab. 3 and graphically in Fig. 8.

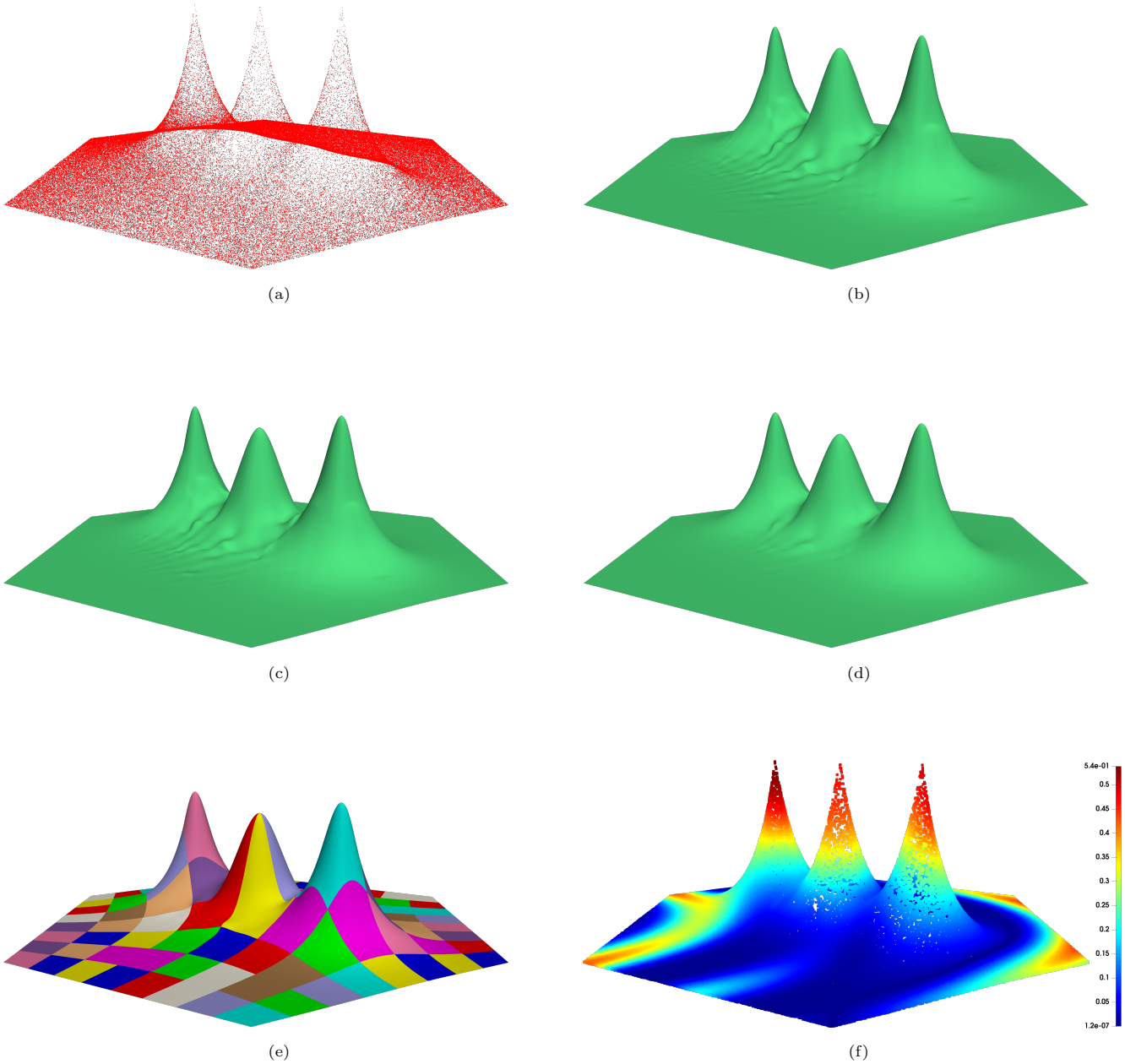
$\boldsymbol{\lambda}$	0	10^{-3}	10^{-2}	10^{-1}
$\boldsymbol{L^\infty}$ **error**	0.164e-00	0.165e-00	0.170e-00	0.196e-00
RMSE	0.308e-02	0.310e-02	0.312e-02	0.375e-02

Table 3: Maximal L^∞ error and RMSE for the surfaces in Example 1 for the surfaces in Example 2 computed making use of 1725 basis functions and progressively bigger smoothing parameter λ. The maximum length of the box containing the model in Fig. 7 is equal to two.

Example 3. In this example, the point cloud is obtained by evaluating a trivariate function defined over the unit sphere $\mathcal{S}^2$. More precisely, the points are sampled from the function

$$f_{\mathcal{S}^2}(x,y,z) = \max\{0, \sin(2\pi x)\sin(2\pi y)\sin(2\pi z)\} + 1\,, \quad (x,y,z)\in\mathcal{S}^2\,. \tag{35}$$

The data we get is composed of 540000 points, while the set of basis functions has been built over a quad mesh, $\mathcal{M}_{\mathcal{S}^2}$, approximating the unit sphere composed of 96 faces. We notice from the error color plot in Fig. 9-(d) that, understandably, the regions in which the error is concentrate involve the peaks present in the point cloud and the lower part of them where there is, in fact, a noticeable slope created by the presence of the max function in (35). Table 4 shows the numerical results for the errors.



(a) (b) (c) (d) (e) (f)

Figure 7: (a): point cloud obtained sampling the function in (34). (b) to (d): approximating surfaces obtained by using the same number of basis function and point cloud with smoothing parameter $\lambda = 0$, $\lambda = 10^{-2}$ and $\lambda = 10^{-1}$, respectively. (e): multipatch coloring of the surface in (d). (f): error color plot of the ℓ_2 distances for the surface in (d). Bounding box with maximum length: 2. Notice how the wrinkles nearby the middle peak diminish when the smoothing parameter increases from (b) to (d).

5.2. Point cloud from ACC surfaces

We provide now fitting examples obtained from big data sets. The point clouds utilized in this section are provided by randomly sampling the Approximate Catmull-Clark surfaces obtained from the construction in [29]. Figs. 10 to 15 show the data we use for our investigation obtained from the ACC surfaces, starting from parameters sampling: the dimension of the clouds goes from a minimum of 549180 to a maximum

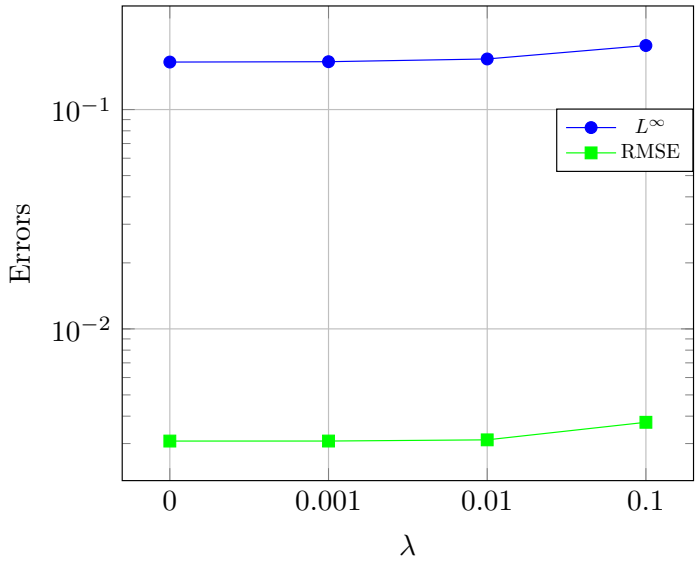
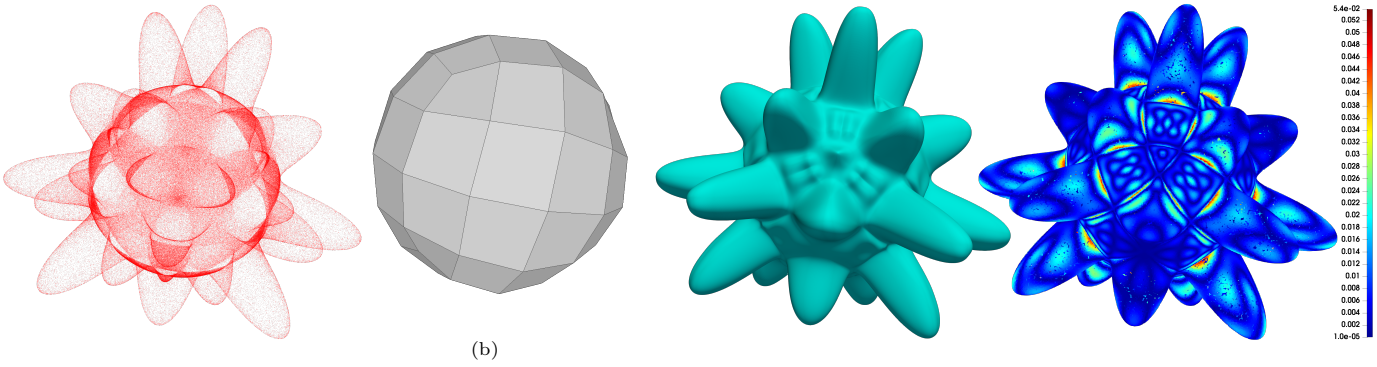


Figure 8: Experimental behavior obtained from the errors represented in Table 3.



(a) (b) (c) (d)

Figure 9: $f_{\mathcal{S}^2}$. (a): point cloud obtained sampling the trivariate function defined in (35) over the unit sphere $\mathcal{S}^2$. (b): quad mesh used to define the G^1 space. (c): final least square surface. (d): error color plot of the ℓ_2 distances. Bounding box with maximum length: 3.

	$\dim\left(\mathbb{G}^1(\mathcal{M}_{\mathcal{S}^2})\right)$	L^∞ error	RMSE
$f_{\mathcal{S}^2}$	1512	0.539e-01	0.107e-01

Table 4: Spline space dimension, maximal error L^∞ and RMSE for the fitting presented in Example 3. The point cloud in Fig. 9 is surrounded by a box with longest size 3.

of 968704 points. In all the experiment presented here we do not consider any smoothing parameter, i.e. $\lambda = 0$, but few iterations of parameter correction [18] are performed to optimize the fitting error; the errors in Table 5 are obtained using the formulas in (32). Since ACC surfaces are not G^1 at the EVs, it is not surprising to see that the maximal fitting error is located around the extraordinary vertices.

5.3. Quadrilateral mesh generation, parametrization and fitting

In the previous experiments, all the point clouds were already equipped with a parametrization; this is because the goal of the paper is to show the quality of the fitting with the basis functions we propose, rather than improving upon parameter computation. Nevertheless, we now present few examples in which both

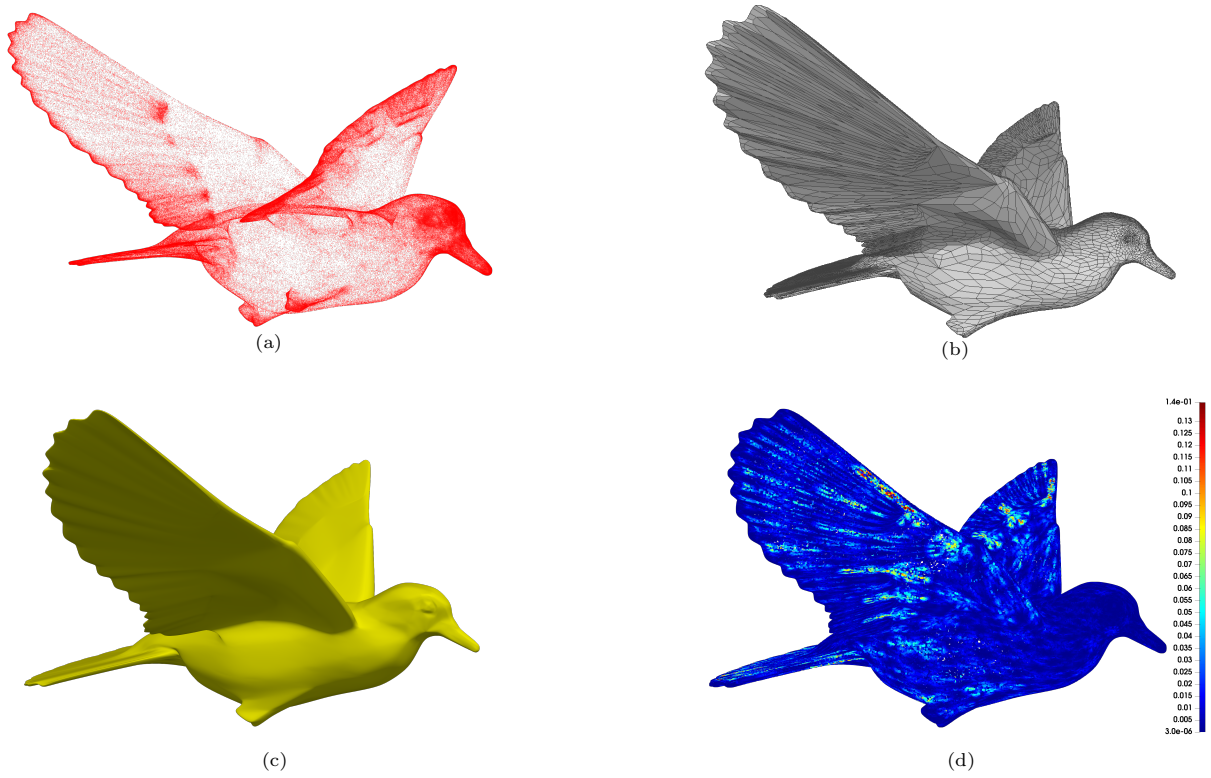


Figure 10: Bird. (a): point cloud obtained sampling an ACC surface. (b): quad mesh used to define the G^1 space. (c): final least square surface. (d): error color plot of the ℓ_2 distances. Bounding box with maximum length: 100.

	bird	**dinosaur**	**hammer**	**hand**	**rabbit**	**venus**
dim $(\mathbb{G}^1(\mathcal{M}))$, n_P	100386, 549180	29885, 680124	48580, 685800	28227, 575424	60220, 968704	55018, 676592
L^∞ **error**	0.138e-00	0.251e-01	0.667e-02	0.417e-01	0.120e-01	0.131e-01
RMSE	0.325e-02	0.348e-03	0.138e-03	0.616e-03	0.302e-03	0.210e-03
3 × parameter correction						
L^∞ **error**	0.754e-01	0.162e-01	0.555e-02	0.320e-01	0.114e-01	0.611e-02
RMSE	0.648e-03	0.205e-03	0.596e-04	0.341e-03	0.145e-03	0.610e-04

Table 5: Maximal error L^∞ and RMSE for the fitting examples presented in Figs. 10 to 15 with a description of the spline space and point cloud features. All the above mentioned models are contained in a box whose longest length is 100.

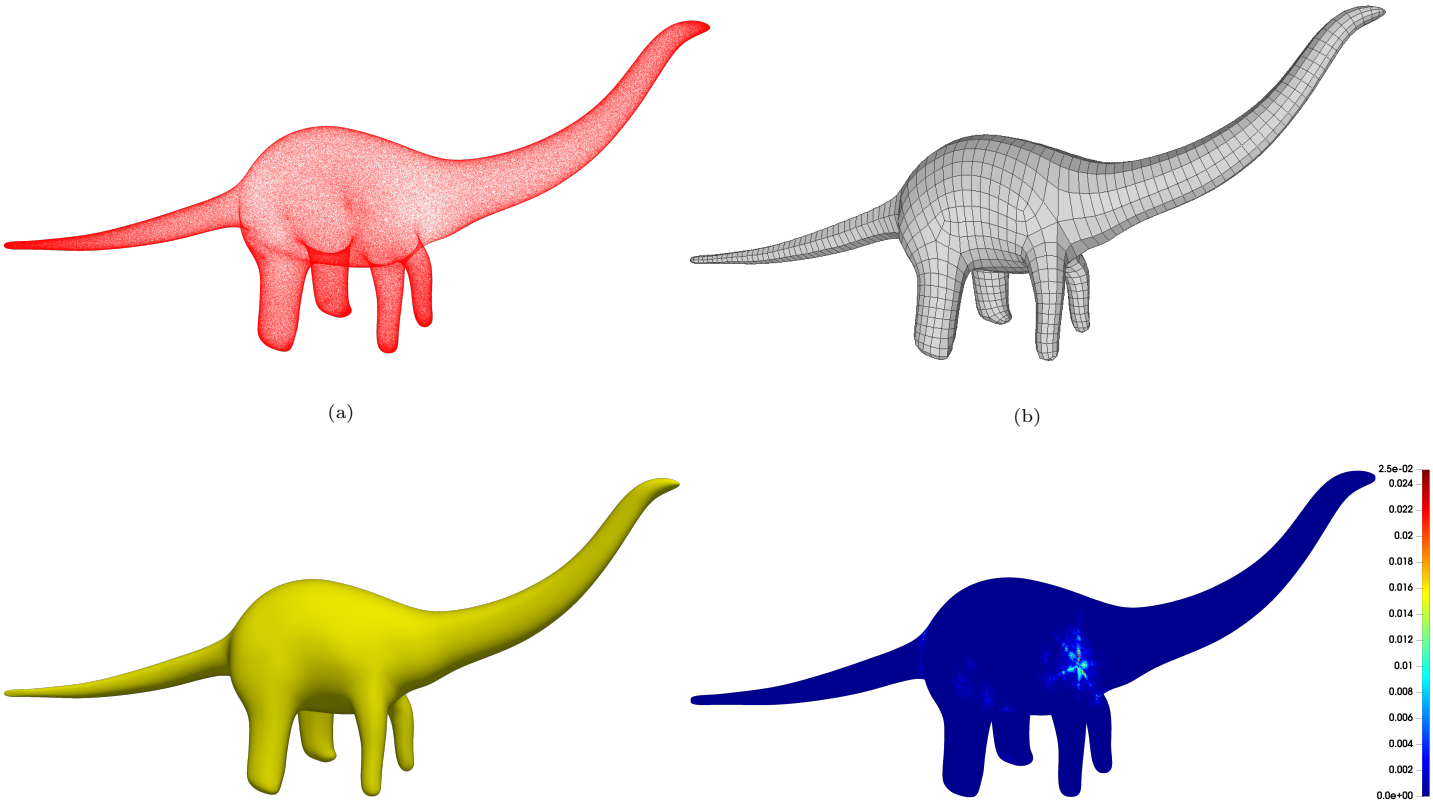
the parametrization of an unorganized point cloud and the construction of a coarse quadrilateral mesh are derived.

Starting from an input point cloud, we first reconstruct a triangular mesh by using the algorithm in [6]; then we apply the so-called 4-8 subdivision scheme [46] to obtain a quadrangular representation of the previous triangular mesh. This procedure leads to a mesh, which is much more refined than we would need for our bases computation. Hence, we coarsen it using Rhino 3D modeler *QuadRemesh* (see [39]), which reduces the faces of the quad mesh without modifying its topology. This quad mesh $\mathcal{M}$ is the mesh on which we define the G^1 space and bases.

In order to associate parameters to a point $P_i = (x_i, y_i, z_i)$ of the point cloud, we compute its orthogonal projection $P_i^\perp = (x_i^\perp, y_i^\perp, z_i^\perp)$ onto the G^1 spline surface $\boldsymbol{G}$, constructed for the quad mesh $\mathcal{M}$ as in [32]. Assume that the point $P_i^\perp$ is laying on the ℓ-th patch of $\boldsymbol{G}$. Let $\boldsymbol{G}^{(\ell)}$ be the geometry map associated with the patch ℓ defined in (30). We define the parameters (u_i, v_i) associated with the point P_i as

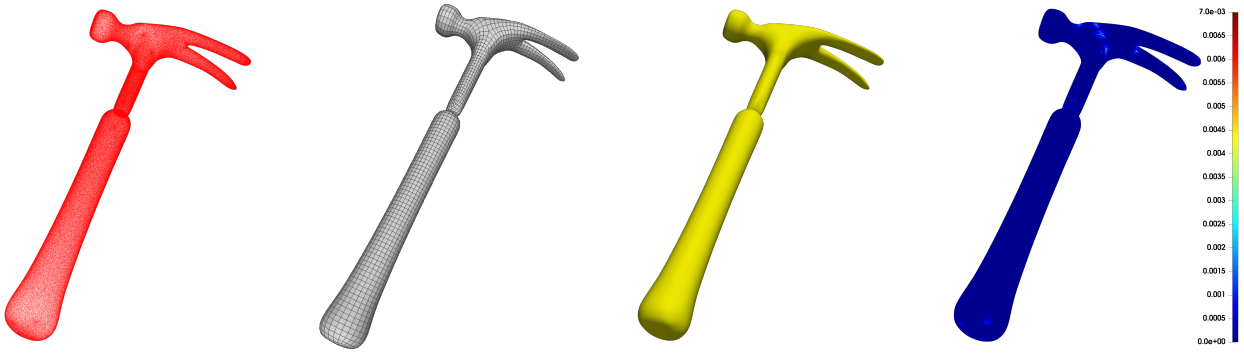
$$(u_i, v_i) = (\boldsymbol{G}^{(\ell)})^{-1}(P_i^\perp)\,;$$

This parameter computation is repeated for all the points of the point cloud.



(a) (b) (c) (d)

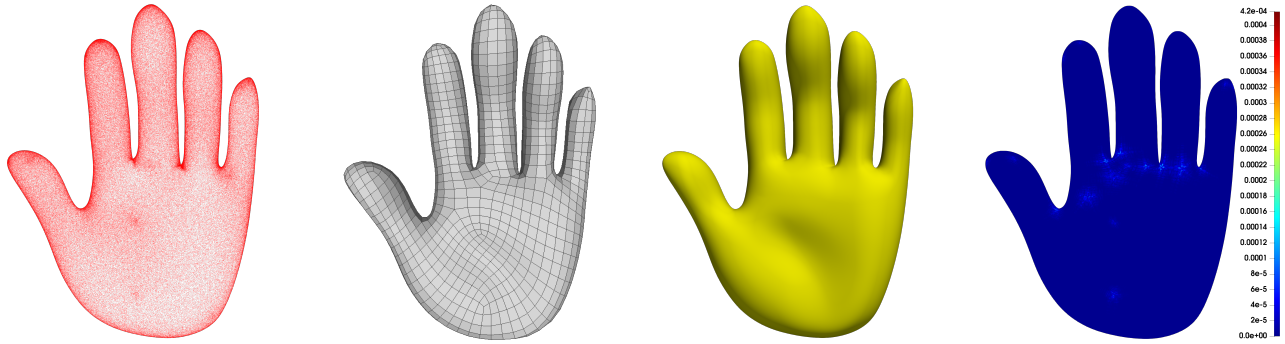
Figure 11: Dinosaur. (a): point cloud obtained sampling an ACC surface. (b): quad mesh used to define the G^1 space. (c): final least square surface. (d): error color plot of the ℓ_2 distances. Bounding box with maximum length: 100.



(a) (b) (c) (d)

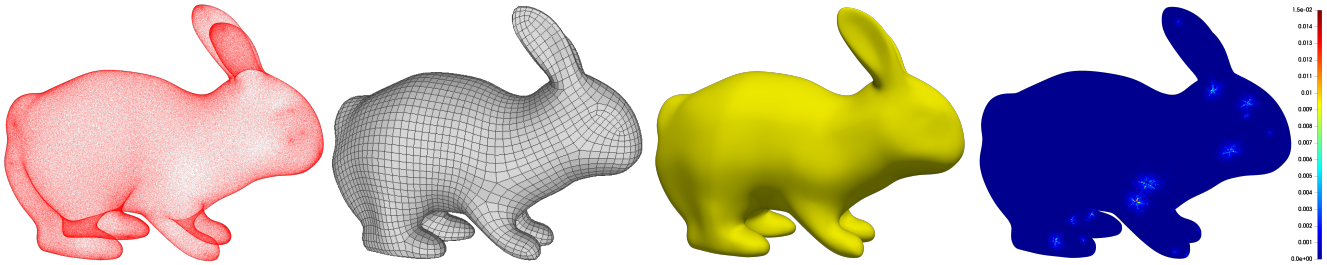
Figure 12: Hammer. (a): point cloud obtained sampling an ACC surface. (b): quad mesh used to define the G^1 space. (c): final least square surface. (d): error color plot of the ℓ_2 distances. Bounding box with maximum length: 100.

Figures 16, 17 and 18 present three examples of point cloud fitting obtained using the pipeline presented in this section. Particular interest goes for the example in Fig. 17, whose initial point cloud presents areas with no points and other with an accumulation of them. Moreover, in all the numerical examples a regularization parameter $\lambda = 10^{-1}$ has been used in order to have a well-posed problem. Table 6 summarizes the key points of the experiments presented in this section.



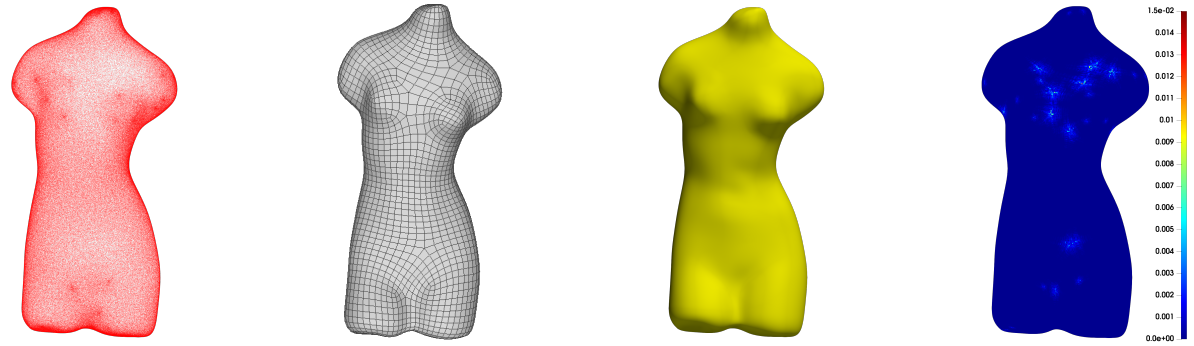
(a) (b) (c) (d)

Figure 13: Hand. (a): point cloud obtained sampling an ACC surface. (b): quad mesh used to define the G^1 space. (c): final least square surface. (d): error color plot of the ℓ_2 distances. Bounding box with maximum length: 100.



(a) (b) (c) (d)

Figure 14: Rabbit. (a): point cloud obtained sampling an ACC surface. (b): quad mesh used to define the G^1 space. (c): final least square surface. (d): error color plot of the ℓ_2 distances. Bounding box with maximum length: 100.

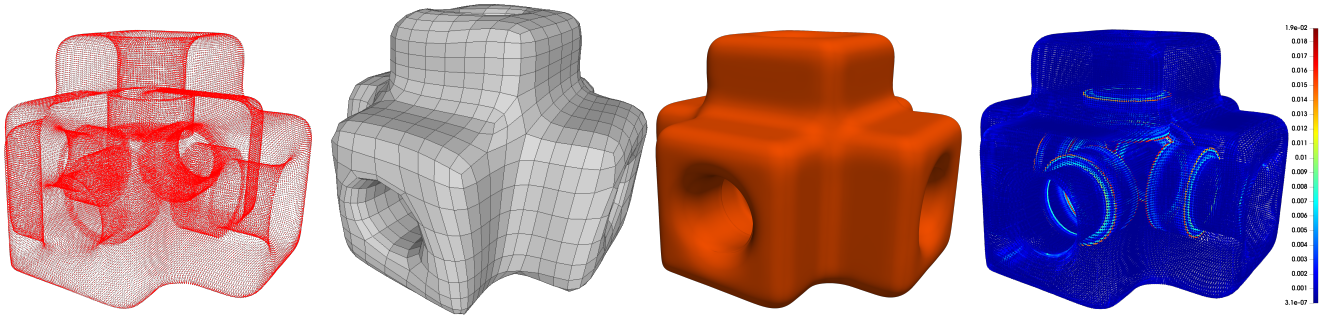


(a) (b) (c) (d)

Figure 15: Venus. (a): point cloud obtained sampling an ACC surface. (b): quad mesh used to define the G^1 space. (c): final least square surface. (d): error color plot of the ℓ_2 distances. Bounding box with maximum length: 100.

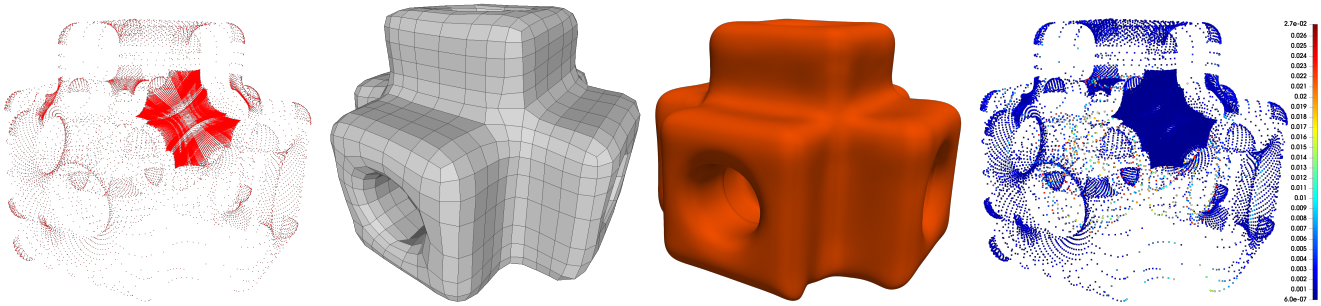
5.4. Comparison with C^0 fitting

Here we present a comparison between surfaces obtained by fitting a point cloud using our proposed G^1 basis functions and using the standard C^0 Bernstein basis on (the faces of) a quad mesh $\mathcal{M}$, which is denoted by $\mathbb{C}^0(\mathcal{M})$. In Table 7 we can notice that, despite the higher dimension of the continuous bases, the errors we get are comparable with the ones obtained from the G^1 construction using fewer basis functions; Figures 19 and 20 present the reflection lines for the treated examples.



(a) (b) (c) (d)

Figure 16: Uniform toy. (a): input point cloud. (b): reconstructed quad mesh used to define the G^1 space. (c): final least square surface. (d): error color plot of the ℓ_2 distances. Bounding box with maximum length: 2.



(a) (b) (c) (d)

Figure 17: Non-uniform toy. (a): input point cloud. (b): reconstructed quad mesh used to define the G^1 space. (c): final least square surface. (d): error color plot of the ℓ_2 distances. Bounding box with maximum length: 2.

	n_P	n_F coarse quad mesh	$\dim\left(\mathbb{G}^1(\mathcal{M})\right)$	L^∞ error	RMSE
Uniform toy	113846	2104	33280	0.190e-01	0.248e-02
Non-uniform toy	30908	1860	29391	0.250e-01	0.277e-02
Icosahedron	208708	7892	125205	0.189e-01	0.142e-02

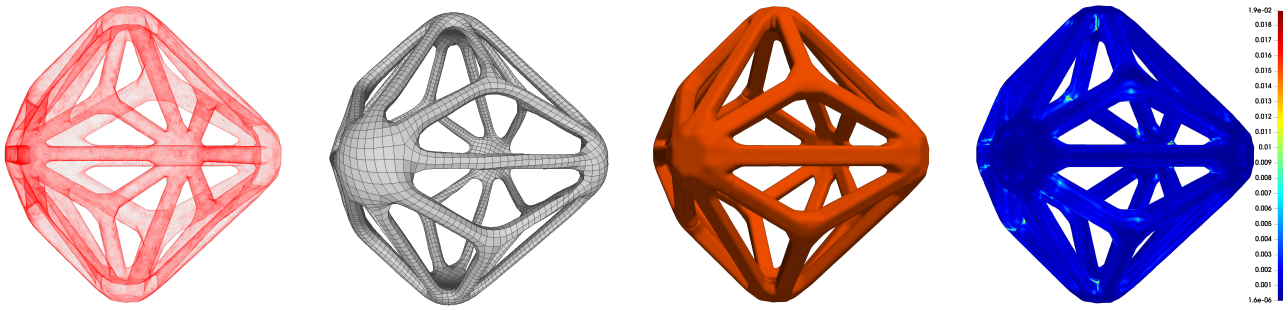
Table 6: Points in the input cloud, quad mesh and corresponding spline space details, maximal error L^∞ and RMSE for the fitting examples in Fig. 16, 17 and 18. All the models are contained in a bounding box whose longest length is 2.

	$\dim\left(\mathbb{C}^0(\mathcal{M})\right)$	$\dim\left(\mathbb{G}^1(\mathcal{M})\right)$	$L^\infty_{C^0}$ error	$L^\infty_{G^1}$ error	RMSE_{C^0}	RMSE_{G^1}
$f_{\mathcal{S}^2}$	2402	1512	0.455e-01	0.539e-01	0.728e-02	0.107e-01
Uniform toy	52602	33280	0.165e-0.1	0.190e-01	0.224e-02	0.248e-02

Table 7: C^0 and G^1 dimensions and errors for the examples in Figure 9 and 16.

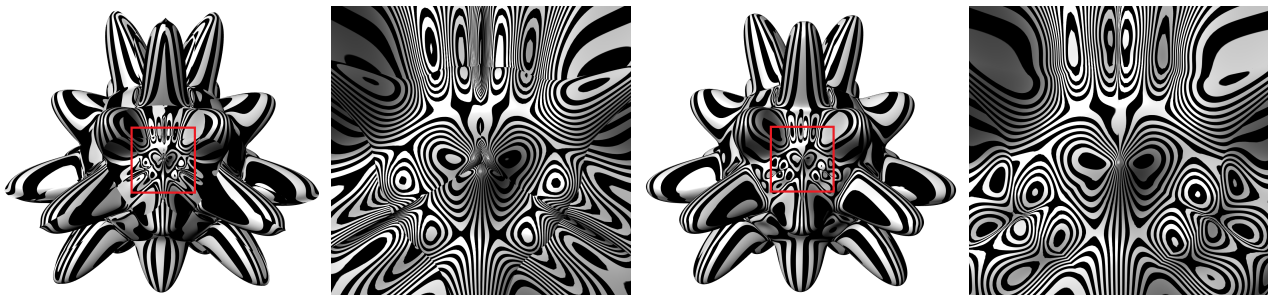
6. Conclusion

In this work we provided an explicit construction of a basis of the space of biquintic G^1 functions over a quad mesh. In particular, the functions of this space are G^1 smooth around extraordinary vertices and at least C^1 elsewhere. The Bézier points defining the multipatch basis functions are explicitly computed starting from the equations defining the space, which are obtained with the use of quadratic gluing data functions defined over the extraordinary edges. Furthermore, we provide an analysis of the resulting basis as well as a combinatorial dimension formula for the G^1 spline space. Finally, several numerical experiments demonstrate the quality of the basis functions in point cloud data fitting problems. These experiments show



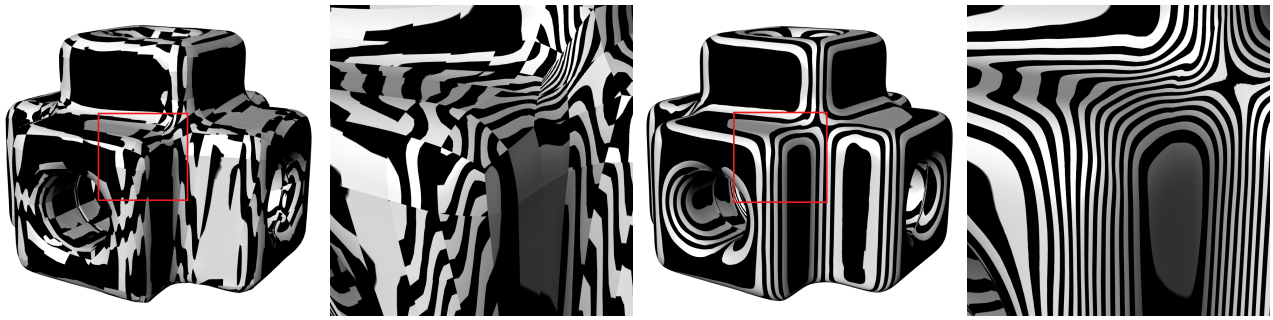
(a) (b) (c) (d)

Figure 18: Icosahedron. (a): input point cloud. (b): reconstructed quad mesh used to define the G^1 space. (c): final least square surface. (d): error color plot of the ℓ_2 distances. Bounding box with maximum length: 2.



(a) (b) (c) (d)

Figure 19: Reflection lines on the C^0 (a) and G^1 (c) surfaces obtained from the point cloud in Fig. 9-(a) and relative zooms, (b) and (d), around an EV.



(a) (b) (c) (d)

Figure 20: Reflection lines on the C^0 (a) and G^1 (c) surfaces obtained from the point cloud in Fig. 16-(a) and relative zooms, (b) and (d), around an EV.

that the overall construction is computationally efficient, even in presence of point clouds with complex geometries, high amount of data or sparse and unequally spaced data.

All the experimental results presented have been obtained using implementations developed in the Julia language for the computation of the basis functions and the G+Smo library (cf. [31]) for the numerical simulations.

An interesting future research is in relation to the isogeometric analysis methodology. More precisely, we would like to investigate and provide an extension of the Bézier bases set presented in this paper to splines patches with inner knots in order to obtain nested spaces of analysis-suitable basis functions to be used in

the numerical solution of partial differential equations.

7. Acknowledgments

This work has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No. 860843.

References

[1] Peter Alfeld and Larry Schumaker. The dimension of bivariate spline spaces of smoothness r for degree $d \geq 4r+1$. *Constr. Approx.*, 3(2):189–197, 1987.
[2] Nina Amenta, Marshall Bern, and Manolis Kamvysselis. A new Voronoi-based surface reconstruction algorithm. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '98*, pages 415–421. ACM Press, 1998.
[3] Philip Azariadis and Nickolas Sapidis. Product design using point-cloud surfaces: a recursive subdivision technique for point parameterization. *Computers in Industry*, 58(8):832–843, 2007.
[4] Michel Bercovier and Tanya Matskewich. *Smooth Bézier Surfaces over Unstructured Quadrilateral Meshes*. Lecture Notes of the Unione Matematica Italiana. Springer International Publishing, 2017.
[5] Matthew Berger, Andrea Tagliasacchi, Lee Seversky, Pierre Alliez, Gael Guennebaud, Joshua Levine, Andrei Sharf, and Claudio Silva. A survey of surface reconstruction from point clouds. *Computer Graphics Forum*, page 27, 2016.
[6] Fausto Bernardini, Joshua Mittleman, Holly Rushmeier, Claudio Silva, and Gabriel Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359, 1999.
[7] Ahmed Blidia, Bernard Mourrain, and Nelly Villamizar. G^1-smooth splines on quad meshes with 4-split macro-patch elements. *Computer Aided Geometric Design*, 52-53:106–125, 2017.
[8] Ahmed Blidia, Bernard Mourrain, and Gang Xu. Geometrically smooth spline bases for data fitting and simulation. *Comput. Aided Geom. Design*, 78:101814, 15, 2020.
[9] Georges-Pierre Bonneau and Stefanie Hahmann. Flexible G^1 interpolation of quad meshes. *Graphical Models*, 76(6):669–681, 2014.
[10] Alexandre Durupt, Sébastien Remy, Guillaume Ducellier, and Benoit Eynard. From a 3d point cloud to an engineering CAD model: a knowledge-product-based approach for reverse engineering. *Virtual and Physical Prototyping*, 3(2):51–59, 2008.
[11] Gerald Farin, Josef Hoschek, and Myung-Soo Kim. *Handbook of computer aided geometric design*. Elsevier, 2002.
[12] Michael S. Floater. Parametrization and smooth approximation of surface triangulations. *Computer Aided Geometric Design*, 14(3):231–250, 1997.
[13] David R. Forsey and Richard H. Bartels. Surface fitting with hierarchical splines. *ACM Transactions on Graphics*, 14(2):134–161, 1995.
[14] Stefanie Hahmann and Georges-Pierre Bonneau. Triangular G^1 interpolation by 4-splitting domain triangles. *Computer Aided Geometric Design*, 17(8):731–757, 2000.
[15] Stefanie Hahmann, Georges-Pierre Bonneau, and Baptiste Caramiaux. Bicubic G^1 interpolation of irregular quad meshes using a 4-split. In *International Conference on Geometric Modeling and Processing*, pages 17–32. Springer, 2008.
[16] Jörg M. Hahn. Geometric continuous patch complexes. *Computer Aided Geometric Design*, 6(1):55–67, 1989-02-01.
[17] Dong Hong. Spaces of bivariate spline functions over triangulation. *Approx. Theory Appl.*, 7(1):56–75, 1991.
[18] Josef Hoschek. Intrinsic parametrization for approximation. *Computer Aided Geometric Design*, 5(1):27–31, 1988.
[19] Mario Kapl, Giancarlo Sangalli, and Thomas Takacs. Dimension and basis construction for analysis-suitable G^1 two-patch parameterizations. *Computer Aided Geometric Design*, 52-53:75–89, 2017.
[20] Mario Kapl, Giancarlo Sangalli, and Thomas Takacs. Isogeometric analysis with C^1 functions on planar, unstructured quadrilateral meshes. *The SMAI Journal of computational mathematics*, S5:67–86, 2019.
[21] Mario Kapl, Giancarlo Sangalli, and Thomas Takacs. An isogeometric C^1 subspace on unstructured multi-patch planar domains. *Computer Aided Geometric Design*, 69:55–75, 2019.
[22] Kęstutis Karčiauskas and Jörg Peters. Biquintic G^2 surfaces via functionals. *Computer Aided Geometric Design*, 33:17–29, 2015.
[23] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics*, 32(3):1–13, 2013.
[24] Gaël Kermarrec, Niklas Schild, and Jan Hartmann. Fitting Terrestrial Laser Scanner Point Clouds with T-Splines: Local Refinement Strategy for Rigid Body Motion. *Remote Sensing*, 13(13):2494, 2021.
[25] Gábor Kiss, Carlotta Giannelli, Urška Zore, Bert Jüttler, David Großmann, and Johannes Barner. Adaptive CAD model (re-)construction with THB-splines. *Graphical Models*, 76(5):273–288, 2014.
[26] Ming-Jun Lai and Larry Schumaker. *Spline functions on triangulations*, volume 110 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 2007.
[27] Xin Li, Jiansong Deng, and Falai Chen. Surface modeling with polynomial splines over hierarchical T-meshes. *The Visual Computer*, 23(12):1027–1033, 2007.
[28] Hongwei Lin, Wei Chen, and Hujun Bao. Adaptive patch-based mesh fitting for reverse engineering. *Computer-Aided Design*, 39(12):1134–1142, 2007.
[29] Charles Loop and Scott Schaefer. Approximating Catmull-Clark subdivision surfaces with bicubic patches. *ACM Transactions on Graphics*, 27(1):8:1–8:11, 2008.
[30] Chenlei Lv, Weisi Lin, and Baoquan Zhao. Voxel structure-based mesh reconstruction from a 3d point cloud. *IEEE Transactions on Multimedia*, 24:1815–1829, 2022.
[31] Angelos Mantzaflaris. An Overview of Geometry Plus Simulation Modules. In *Mathematical Aspects of Computer and Information Sciences*, Lecture Notes in Computer Science, pages 453–456, Cham, 2020. Springer International Publishing.

[32] Michelangelo Marsala, Angelos Mantzaflaris, and Bernard Mourrain. G^1 – smooth biquintic approximation of Catmull-Clark subdivision surfaces. *Computer Aided Geometric Design*, 99:102158, 2022.
[33] Bernard Mourrain, Raimundas Vidunas, and Nelly Villamizar. Geometrically continuous splines for surfaces of arbitrary topology. *Computer Aided Geometric Design*, 45:108–133, 2016.
[34] Bernard Mourrain and Nelly Villamizar. Homological techniques for the analysis of the dimension of triangular spline spaces. *J. Symbolic Comput.*, 50:564–577, 2013.
[35] Shayan Nikoohemat, Abdoulaye A. Diakité, Sisi Zlatanova, and George Vosselman. Indoor 3d reconstruction from point clouds for optimal routing in complex buildings to support disaster management. *Automation in Construction*, 113:103109, 2020.
[36] Jörg Peters and Ulrich Reif. *Subdivision surfaces*. Springer, 2008.
[37] Helmut Pottmann, Stefan Leopoldseder, and Michael Hofer. Approximation with active B-spline curves and surfaces. In *10th Pacific Conference on Computer Graphics and Applications, 2002. Proceedings.*, pages 8–25, 2002.
[38] Florent Poux, Roland Billen, Jean-Paul Kasprzyk, Pierre-Henri Lefebvre, and Pierre Hallot. A built heritage information system based on point cloud data: HIS-PC. *ISPRS International Journal of Geo-Information*, 9(10):588, 2020.
[39] Rhino3D. https://rhino3d.co.uk/rhino-for-windows/an-introduction-to-quadremesh-in-rhino3d-v7/.
[40] Larry Schumaker. Bounds on the dimension of spaces of multivariate piecewise polynomials. *Rocky Mountain J. Math.*, 14(1):251–264, 1984.
[41] Xiquan Shi, Tianjun Wang, and Piqiang Yu. A practical construction of G^1 smooth biquintic B-spline surfaces over arbitrary topology. *Computer-Aided Design*, 36(5):413–424, 2004.
[42] Martin Sinko, Patrik Kamencay, Robert Hudec, and Miroslav Benco. 3d registration of the point cloud data using ICP algorithm in medical image analysis. In *2018 ELEKTRO*, pages 1–6, 2018-05.
[43] Arkadiusz Sitek, Ronald H. Huesman, and Grant T. Gullberg. Tomographic reconstruction using an adaptive tetrahedral mesh defined by a point cloud. *IEEE Transactions on Medical Imaging*, 25(9):1172–1179, 2006-09.
[44] George Sithole and George Vosselman. Experimental comparison of filter algorithms for bare-earth extraction from airborne laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 59(1):85–101, 2004-08-01.
[45] Vibeke Skytt, Gaël Kermarrec, and Tor Dokken. LR B-splines to approximate bathymetry datasets: an improved statistical criterion to judge the goodness of fit. *International Journal of Applied Earth Observation and Geoinformation*, 112:102894, 2022.
[46] Luiz Velho and Denis Zorin. 4–8 subdivision. *Computer Aided Geometric Design*, 18(5):397–427, 2001. Subdivision Algorithms.
[47] Qian Wang and Min-Koo Kim. Applications of 3d point cloud data in the construction industry: a fifteen-year review from 2004 to 2018. *Advanced Engineering Informatics*, 39:306–319, 2019.
[48] Wenping Wang, Helmut Pottmann, and Yang Liu. Fitting B-spline curves to point clouds by curvature-based squared distance minimization. *ACM Trans. Graph.*, 25:214–238, 2006.
[49] Yan Wang, Zixian Fan, and Yingchun You. Application research of earth volume calculation based on 3d laser point cloud data. *IOP Conference Series: Materials Science and Engineering*, 780(3):032050, 2020-03.
[50] Ziwei Wang, Sijie Yan, Long Wu, Xiaojian Zhang, and BinJiang Chen. Robust point clouds registration with point-to-point l_p distance constraints in large-scale metrology. *ISPRS Journal of Photogrammetry and Remote Sensing*, 189:23–35, 2022-07-01.
[51] Pete J. Watt and Daniel N.M. Donoghue. Measuring forest structure with terrestrial laser scanning. *International Journal of Remote Sensing*, 26(7):1437–1446, 2005.
[52] Volker Weiss, Làszló. Andor, Gabor Renner, and Tamas Várady. Advanced surface fitting techniques. *Computer Aided Geometric Design*, 19(1):19–42, 2002.
[53] Linfu Xie, Han Hu, Qing Zhu, Xiaoming Li, Shengjun Tang, You Li, Renzhong Guo, Yeting Zhang, and Weixi Wang. Combined rule-based and hypothesis-based method for building model reconstruction from photogrammetric point clouds. *Remote Sensing*, 13(6):1107, 2021-01.
[54] Jianing Yao, Alexander Anderson, and Jannick P. Rolland. Point-cloud noncontact metrology of freeform optical surfaces. *Optics Express*, 26(8):10242–10265, 2018-04-16.
[55] Xiuyang Zhao, Caiming Zhang, Li Xu, Bo Yang, and Zhiquan Feng. IGA-based point cloud fitting using B-spline surfaces for reverse engineering. *Information Sciences*, 245:276–289, 2013-10-01.
[56] Zaiping Zhu, Andres Iglesias, Lihua You, and Jian Jun Zhang. A review of 3d point clouds parameterization methods. In *Computational Science–ICCS 2022: 22nd International Conference, London, UK, June 21–23, 2022, Proceedings, Part III*, pages 690–703. Springer, 2022.