

Understanding hierarchical B-splines with a case study: approximation of point clouds from TLS observations

Gaël Kermarrec, Jan Hartmann, Hannes Faust, Knut Hartmann, Reza Besharat, Gad Samuel, Chen Lixian and Hamza Alkhatib

Summary

Static or kinematic Terrestrial Laser Scanners (TLS) are a way to acquire all kinds of scenes or objects, offering the possibility to collect millions of 3D points in a short time. The corresponding point clouds (PC) can be analysed in standard software or mathematically approximated. Regression B-spline surfaces allow great flexibility to model PC, since no predetermined geometric primitives such as circles, planes or cylinders restrict the fitting. These surfaces are defined by a tensor product of B-spline basis functions, which limit the local refinement. This drawback can be circumventing by using hierarchical B-splines. In this contribution, we propose to explain in a didactical way the mathematical concepts and apply hierarchical B-splines approximation to TLS observations from a bridge under load. This example highlights the high potential of hierarchical refinement to gain a finer local control over the surface to approximate. It paves the way to model PC from TLS observations and performs rigorous test for deformation based on a fine mathematical approximation of the surface.

Zusammenfassung

Statische oder kinematische terrestrische Laserscanner (TLS) bieten die Möglichkeit, alle Arten von Szenen oder Objekten zu erfassen und in kurzer Zeit Millionen von 3D-Punkten zu sammeln. Die entsprechenden Punktwolken können in Standardsoftware analysiert oder mathematisch approximiert werden. Regressions-B-Spline-Flächen ermöglichen eine große Flexibilität beim Modellieren von TLS Beobachtungen, da keine vorgegebenen geometrischen Grundelemente wie Kreise, Ebenen oder Zylinder die Anpassung einschränken. Diese Oberflächen werden durch ein Tensorprodukt von B-Spline-Basisfunktionen definiert, die die lokale Verfeinerung begrenzen. Dieser Nachteil kann umgangen werden, indem hierarchische B-Splines verwendet werden. In diesem Beitrag schlagen wir vor, die mathematischen Konzepte auf didaktische Weise zu erklären und hierarchische B-Splines-Approximation auf TLS-Beobachtungen von einer Brücke unter Last anzuwenden. Dieses Beispiel zeigt das große Potenzial hierarchischer Verfeinerung, um eine genauere lokale Kontrolle über die Oberfläche zu erhalten. Es ebnet den Weg für die Modellierung von PC anhand von TLS-Beobachtungen und führt strenge Deformationstests auf der Grundlage einer feinen mathematischen Approximation der Oberfläche durch.

Keywords: Terrestrial Laser Scanner, hierarchical B-splines, surface approximation, deformation analysis, knot refinement, B-spline surfaces

1 Introduction

The fitting of curves or surfaces through a set of observations appears in many disciplines from mathematics, engineering up to medical imaging. Mathematically, the parameters of a model that minimizes an error criterion have to be determined given a set of noisy and often unequally spaced data values.

Because splines (de Boor 2001) have interesting properties such as their closed form expressions, the recursion relations and polynomial based basis functions, they have enjoyed special attention, particularly in computer aided geometric design and computer graphics to smooth or interpolate observations (Unser 1997). Splines are piecewise polynomials of a certain degree, with continuity constraints in the knots that connect the polynomial pieces. They are formally defined by a recurrence relation, which allows local variation of the degree of continuity. The flexibility of the approximation of point clouds (PC) with B-spline curves or surfaces arises from its independence to pre-defined geometric primitives, such as circle, ellipsoid, plane. They satisfy the strong convex hull property and have a fine shape control: so-called control points (CP) define the control polygon of the spline, which is a rough sketch of the curve itself. By simply manipulating the control net, the shape of the curve or surfaces can be changed locally.

When reconstructing an object starting with some discrete observations either an interpolation or an approximation can be performed (Lee 2000). For noisy observations, due, e.g., to measurement errors, the latter is preferable; the distance between the points and its mathematical approximation is often minimized in least-squares sense (Aigner and Jüttner 2009). The coordinates of the CP are the parameters to be estimated. They are weighted by the B-spline functions evaluated at the pre-defined parameters values of the PC.

B-spline or NURBS (Non-uniform rational basis spline) curves (or surfaces) gain an increased interest in engineering geodesy to approximate PC from, e.g., Terrestrial Laser Scanner (TLS). Interested readers should refer to Koch (2009), Bureick et al. (2016). One of the most promising application in geodesy is to use surface approximation for deformation analysis (Kermarrec et al. 2019, Harmening and Neuner 2020). Indeed, B-spline surfaces allow a parametric representation of the point clouds leading to:

- a reduction of the computational burden associated with the high number of recorded points

- a promising way to derive rigorous testing strategy for deformation based on the estimated parameters (the CP)

Unfortunately, TLS observations are scattered and unequally spaced, with a high varying density of the data distribution. When approximated with B-spline surfaces, the tensor-product approach precludes localized refinement: surface reconstruction using a global least-squares (LS) approach can become rapidly computationally demanding as the number of parameters increases (Rabut 2005). Unwanted oscillations in the reconstructed surface may occur (Bracco et al. 2017) due to the high number of degrees of freedom with respect to the local number of data. On the contrary, when less parameters are estimated, local details may be lost due to the low-pass filtering effect of the B-splines basis (Goshtasby et al. 1990). Important information about potential deformation when comparing the approximation at two epochs may be lost. Introduced by Forsey and Bartels (1988), the hierarchical B-splines provides a framework to overcome this drawback. They were further improved by Kraft (1997), who developed a strategy to complement the mesh refinement procedure and are mainly used in isogeometric analysis (Vuong et al. 2011). Extension of local LS approximations to hierarchical spline spaces were addressed in Bracco et al. (2016) and alternatives using variable spline degree proposed in Costantini (2000) or Kjetil et al. (2015). In this contribution, we propose to explain didactically the concept of hierarchical B-splines and show how they can be used in geodesy as an alternative to the global approach. To that aim, we will use both simulated TLS point clouds and a real data set analysis to highlight the potential of the framework for local deformation analysis.

The remainder of this contribution is as follows: in a first part, we introduce the mathematical concept of hierarchical B-splines. A second section is devoted to simulations, whereas the last section presents the results obtained by using real data from a bridge under load. We conclude this contribution by summarizing the advantages of using hierarchical refinement with respect to global approximation.

2 Mathematical background

In this section, we introduce shortly the main concepts of B-spline curves and surfaces. Interested readers should refer to Cohen et al. (2001) for an introduction to geometric modelling with splines. In this section, we will highlight the drawback associated with the tensor product approach. This is a precondition to understand why a hierarchical mesh refinement is needed.

For didactical purpose, the way to construct the hierarchical basis will be explained both in 1D and 2D.

2.1 B-spline curves

The disadvantages associated with Bezier curves (lack of local control, high degree curves needed for complex shapes, see Piegl and Tiller (1997), can be addressed by piecewise polynomial curves called splines defined over a B-spline basis.

In that case, the curve $C(u)$ is defined as:

$$C(u) = \sum_{i=1}^{n_{CP}} B_{i,p}(u) p_i, \quad (1)$$

where p_i is the i^{th} CP from a total of n_{CP} and $B_{i,p} = B_{i,p,U}$ the B-spline function of degree p depending on the non-decreasing sequence of real numbers called knots:

$$U = (u_i)_{i=0}^{n_{CP}+p+2}. \quad (2)$$

We call knot span $[u_i, u_{i+1}]$ the half open interval and U the knot vector. $B_{i,p}$ is evaluated at a given parameter value u using the recurrence relation (de Boor 1972):

$$B_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} B_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} B_{i+1,p-1}(u) \quad (3)$$

starting with $B_{i,0}(u) = 1$ if $u \in [u_i, u_{i+1}]$ and 0 otherwise, with the convention that anything divided by zero is zero.

Parametric B-spline curves are defined by letting the control points be points in \mathbb{R}^2 instead of real numbers, so that the obtained curve takes values in \mathbb{R}^2 as well (Zhao et al. 2019). A uniform, cord length or centripetal parametrisation can be used (Piegl and Tiller 1997). The parametrisation may impact the curve fitting for challenging geometries (Harmening and Neuner 2015).

In Fig. 1 (top right), the B-splines basis made of all B-spline functions is shown together with the obtained curve for the given knot vector (Fig. 1 left). The position of the CP influences the shape of only few polynomial segments of the whole curve. This is illustrated in Fig. 1 (middle left) for which one CP was moved, i.e. this is why B-splines are said to provide local support. By introducing new knots in the knot vector, it is moreover possible to model sharp edges (see Fig. 1 bottom for which one knot was introduced between 0.3 and 0.4 compared to Fig. 1 top). A knot multiplicity, which corresponds to a case where a knot is repeated at least two times in the knot vector – for instance $[..., 0.2 \ 0.2 \ 0.3, ...]$ – allows to introduce discontinuities or break-lines. Please note that the three times 0 and three times 1 are needed to clamp the curve to the first and last control point (Piegl and Tiller 1997).

2.1.1 Knot insertion

As the name suggests, knot insertion is related to the insertion of knots into an existing knot vector (Equation 2). Automatically, the new spline space is refined and

contains more B-splines, increasing the flexibility to model more complicated curves. More specifically, any curve in the original space can also be represented in terms of the B-splines in the refined space: the control polygon defined by the control points is only moved closer to the spline itself without changing the shape of the curve (see Fig. 1 bottom left) for an illustration of that property.

After adding a new knot, the value of $m = n_{CP} + p + 2$ is increased by one. By inserting a new knot, a new control point is automatically added, or better said, the control polygon is “remixed” (see Fig. 1 bottom).

Mathematically, the knot insertion is performed as follows:

1. We state that the new knot is given by $\bar{u} \in [u_j, u_{j+1}]$.
2. The knot vector becomes thus

$$\bar{\mathbf{U}} = [\bar{u}_0 = u_0, \dots, \bar{u}_j = u_j, \bar{u}_{j+1} = \bar{u}, \bar{u}_{j+2} = u_{j+1}, \dots, \bar{u}_{m+1} = u_m].$$

If the new knot $\bar{u} \in [u_j, u_{j+1}]$ is equal to u_j , the multiplicity of u_j is increased by one.

3. Based on $\bar{\mathbf{U}}$, the new curve is defined as:

$$C(u) = \sum_{i=1}^{n_{CP}+1} \bar{B}_{i,p}(u) \bar{p}_i \quad (4)$$

with $\bar{B}_{i,p} = \bar{B}_{i,p,\bar{\mathbf{U}}}$. \bar{p}_i is the new i^{th} CP from a total of $n_{CP} + 1$.

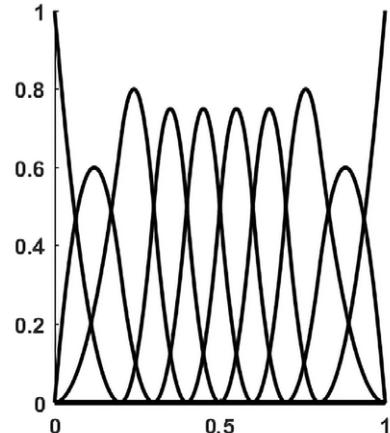
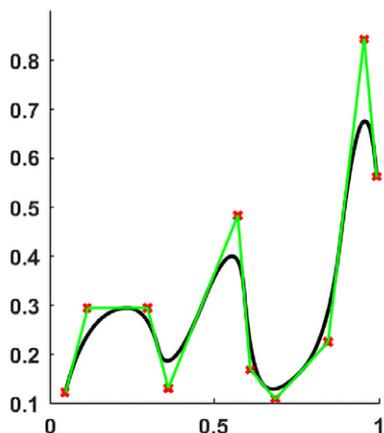
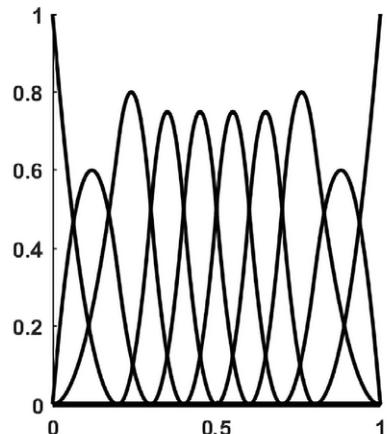
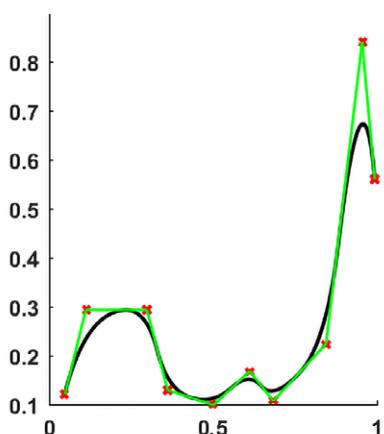
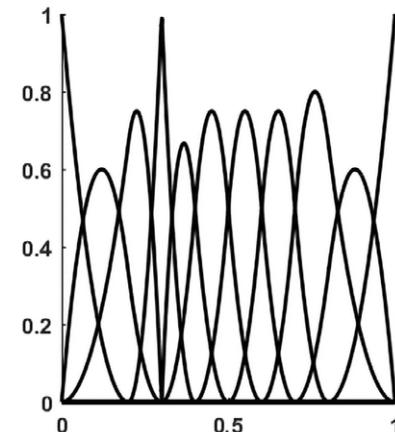
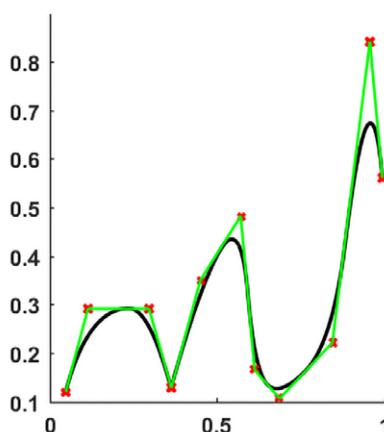


Fig. 1:
Top: Left: B-spline curve with its control polygon (green line) and the corresponding control points (red cross). Right: The associated B-spline basis functions. The knot vector sequence is [0, 0, 0, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 1, 1, 1], grad 3.



Middle: Left: B-spline curve with its control polygon after moving one CP. Right: The corresponding B-spline basis functions. The knot vector sequence is the same as Fig. 1 top.



Bottom: Left: B-spline curve with its control polygon after adding one knot. Right: The corresponding B-spline basis functions. The knot vector sequence is [0, 0, 0, 0.2, 0.3, 0.35, 0.4, 0.5, 0.6, 0.7, 0.8, 1, 1, 1].

The new coefficient can be obtained by the simple linear relationship (Piegl and Tiller 1997):

$$\bar{p}_i = \alpha_i p_i + (1 - \alpha_i) p_{i-1}$$

$$\text{where } \alpha_i = \begin{cases} 1, & \text{if } i \leq j-p \\ \frac{\bar{u} - u_i}{u_{i+p} - u_i}, & \text{if } j-p+1 \leq i \leq j \\ 0, & \text{if } j+1 \leq i \end{cases} \quad (5)$$

The new CP in the refined space can be expressed using the transfer matrix \mathbf{R} : $\bar{\mathbf{P}} = \mathbf{R}\mathbf{P}$, with $\mathbf{P}, \bar{\mathbf{P}}$ the matrices of the CP in the original and refined spaces, respectively and

$$\mathbf{R} = \begin{bmatrix} \alpha_0 & 0 & 0 & \dots & \dots & 0 \\ 1 - \alpha_1 & \alpha_1 & 0 & & & 0 \\ 0 & 1 - \alpha_2 & \alpha_2 & \dots & \dots & 0 \\ \dots & \dots & \dots & & & \\ 0 & 0 & 0 & 1 - \alpha_{n-1} & \alpha_{n-1} & \\ 0 & 0 & 0 & \dots & 0 & 1 - \alpha_n \end{bmatrix}. \quad (6)$$

Similarly, the new basis of B-splines functions is given by $\bar{\mathbf{B}} = \mathbf{R}\mathbf{B}$, where the B-spline basis \mathbf{B} is the collection of all $B_{i,p,\bar{\mathbf{U}}}$ and $\bar{\mathbf{B}}$ of all $B_{i,p,\bar{\mathbf{U}}}$. Please refer to D'Angella et al. (2018) for a detailed example on how \mathbf{R} is built.

2.1.2 Hierarchical B-splines 1D

We start from an initial knot vector \mathbf{U}_0 – often taken equidistant for the sake of simplicity. This knot vector is called “of level 0” or the “coarse knot vector”. We refine this vector, keeping the equidistant spacing, but adding new knots in the middle of the knot span \mathbf{U}_0 . The finer knot vector is called \mathbf{U}_1 “of level 1”. From now, rather than mathematical formulas, and for the sake of simplicity, we use a didactical example to further explain the construction of the hierarchical basis.

Example hierarchical basis 1D

We take two basis – here with degree 2 – in Fig. 2 with a knot vector $\mathbf{U}_0 = [0, 0, 0, \frac{1}{8}, \frac{2}{8}, \dots, \frac{7}{8}, 1, 1, 1]$ and $\mathbf{U}_1 = [0, 0, 0, \frac{1}{16}, \frac{1}{8}, \dots, \frac{15}{16}, 1, 1, 1]$.

Mathematically, we wish to refine the green area Ω^1 , a subset of Ω^0 , which we call the parameter domain: $\Omega^1 \subset \Omega^0$. The idea is simple: we take all functions B_{i,p,\mathbf{U}_1} which are contained in Ω^1 only. In other words, and as illustrated by Fig. 3 (bottom), the last two functions should be 0 exactly at “the boundary” of the green area. In a second step, we delete the one that are not in Ω^1 . The remaining functions are the “active B-splines of level 1”. Similarly, we keep only the active B-splines of level 0

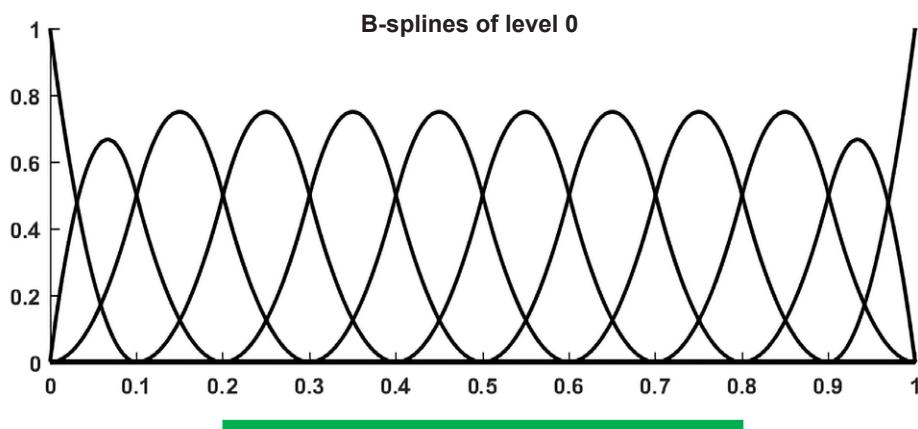
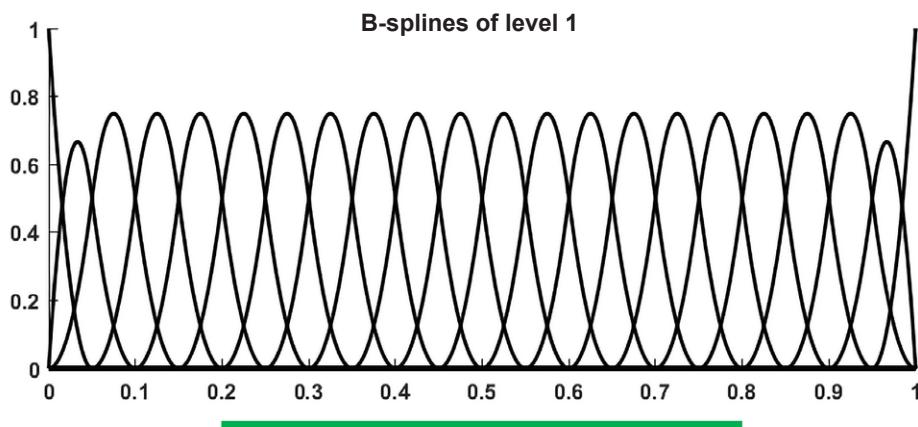


Fig. 2:
Top B-spline basis level 0.
The knot vector is equidistant
[0 0 0 0.1 0.2 0.3 0.4 0.5 0.6
0.7 0.8 0.9 1 1 1].



Bottom: B-spline basis level 1.
The knot vector is [0 0 0 0.05
0.1 0.15 0.2 0.25 0.3 0.35
0.4 0.45 0.5 0.55 0.6 0.65
0.7 0.75 0.8 0.85 0.9 0.95
1 1 1].

to complete the refined basis (Fig. 3 top). We delete the functions of level 0 from the one starting (i.e. being non-0) at the green area to the one ending (i.e. being 0) at the other end of the subset. The obtained hierarchical spline space is illustrated in Fig. 4. Clearly, the partition of unity is lost since the sum of the B-splines functions can exceed 1: $\sum_{i=1}^{n_{cp}} B_{i,p}(u) \neq 1 \Omega^1$.

Moreover, there can be large overlaps of the supports of the basis functions. Such drawbacks can be addressed within the context of truncated B-splines (TBH, Gianelli et al. 2012).

2.2 B-spline surfaces

2.2.1 The tensor product

B-spline surfaces are computed by taking the tensor-product of two univariate B-spline functions defined over two corresponding knot vectors \mathbf{U}^1 and \mathbf{U}^2 of size n^1 and n^2 , respectively.

In mathematics, a tensor refers to objects that have multiple indices (Bourbaki 1989). Roughly speaking this can be thought of as a multidimensional array or a way of building a new vector space from old vector spaces.

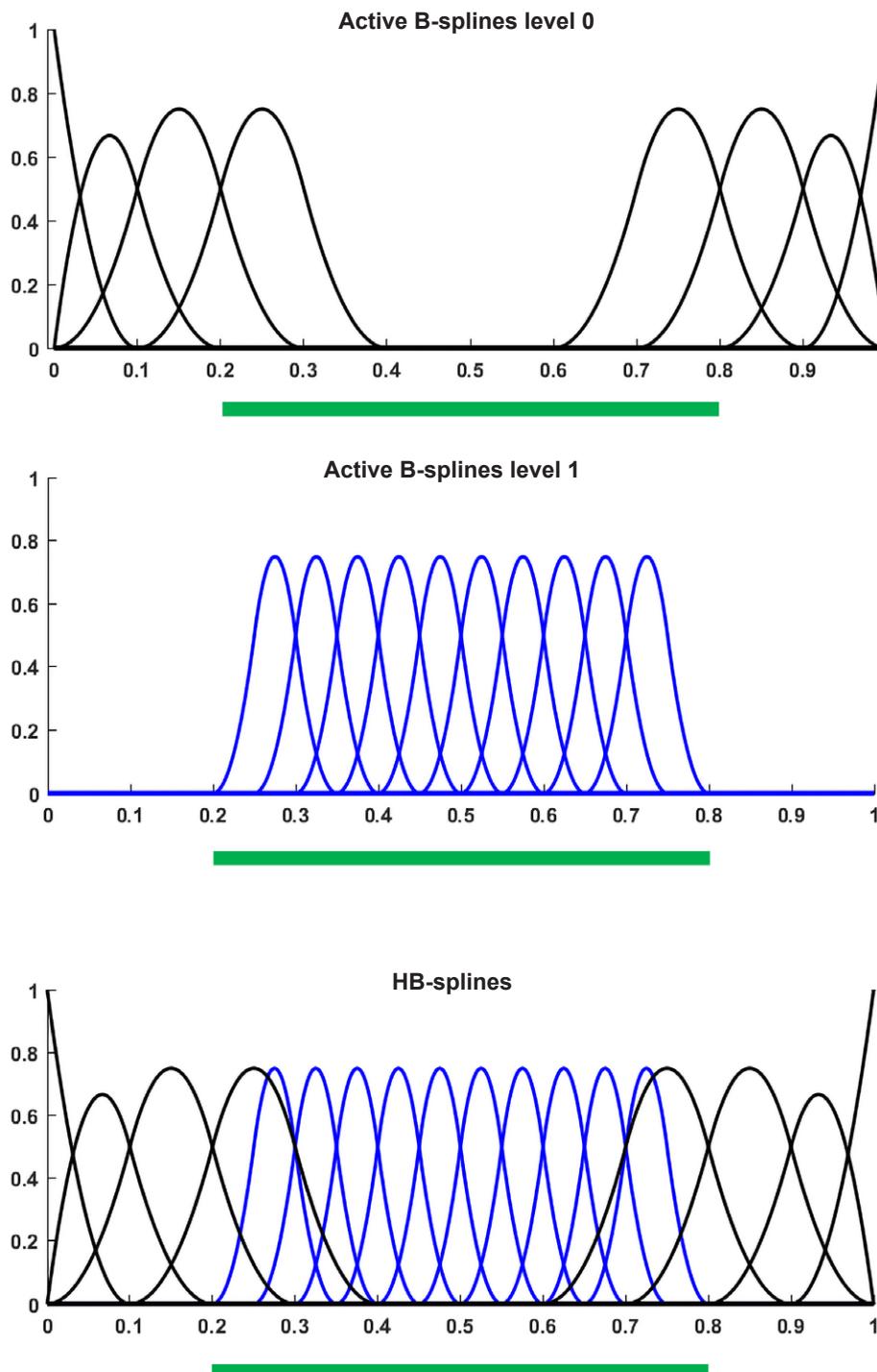


Fig. 3:
Top: Active B-splines level 0.
Bottom: Active B-splines level 1. The subdivision interval is the green rectangle.

Bottom: B-spline basis level 1.
The knot vector is [0 0 0 0.05
0.1 0.15 0.2 0.25 0.3 0.35
0.4 0.45 0.5 0.55 0.6 0.65
0.7 0.75 0.8 0.85 0.9 0.95 1
1 1].

Fig. 4:
The hierarchical splines corresponding to a refinement on the green subdomain

The tensor product of vectors is a matrix, as this short numerical example shows:

$$\mathbf{v} \otimes \mathbf{w} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \otimes \begin{bmatrix} 4 & 5 \end{bmatrix} = \begin{bmatrix} 1 \cdot 4 & 1 \cdot 5 \\ 2 \cdot 4 & 2 \cdot 5 \\ 3 \cdot 4 & 3 \cdot 5 \end{bmatrix} = \begin{bmatrix} 4 & 5 \\ 8 & 10 \\ 12 & 15 \end{bmatrix}. \quad (7)$$

Consecutively, by multiplying two B-spline functions, a matrix is obtained, which can be represented – in 2D and for cubic B-splines – by a Gaussian bell curve. Fig. 5 shows the results of the tensor product between two B-spline functions.

More precisely, the B-spline surface will have the form:

$$S(u^1, u^2) = \sum_{i=0}^{n^1} \sum_{j=0}^{n^2} B_{i, p^1, \mathbf{U}^1}(u^1) B_{j, p^2, \mathbf{U}^2}(u^2) p_{ij}, \quad (8)$$

$\mathbf{U}^1 = (u_i^1)_{i=0}^{n^1+p^1+2}$, $\mathbf{U}^2 = (u_j^2)_{j=1}^{n^2+p^2+2}$ are the knot vectors associated with the B-spline functions in the two parameter directions u^1, u^2 , respectively, with degree p^1, p^2 . There are $n^1 + 1$ and $n^2 + 1$ B-spline functions to be evaluated in the two directions and p_{ij} are the CP. In 2D, they define a control mesh, as a generalization of the 1D control polygon.

The tensor product of the 1D B-splines forms a mesh (also called grid), which we consider as rectangular: this is, by definition, a set of points and cells that forms a network. The size of the cells depends on the knot placement. In the following, we will call mesh refinement a new mesh based on the addition of cells or points, reducing the local characteristic length of edges. Fig. 6 shows an example of a rectangular mesh. A simple mesh refinement subdivides the cells dyadically, each refined cell being partitioned in 4 new one. This technique can increase local mesh resolution, but also increase the number of parameters to be estimated (here the number of CP), by increasing the number of degrees of freedom of the system.

A way to avoid this explosion of number of CP to estimate is to intend to insert knots only locally. As for the univariate case, this leads to a set of new control points

$$\bar{p}_{i,j} = (\otimes \mathbf{R}^k) p_{i,j}, k=1,2, \quad (9)$$

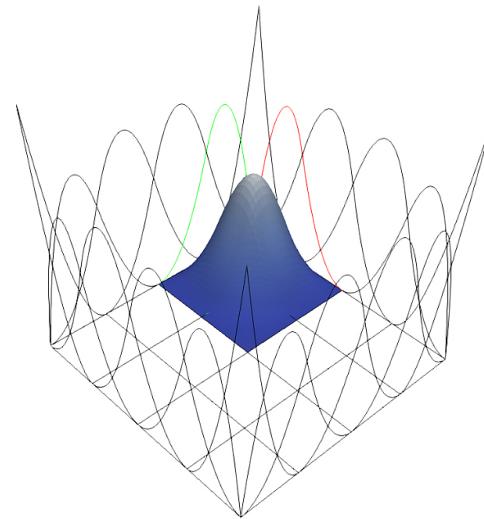
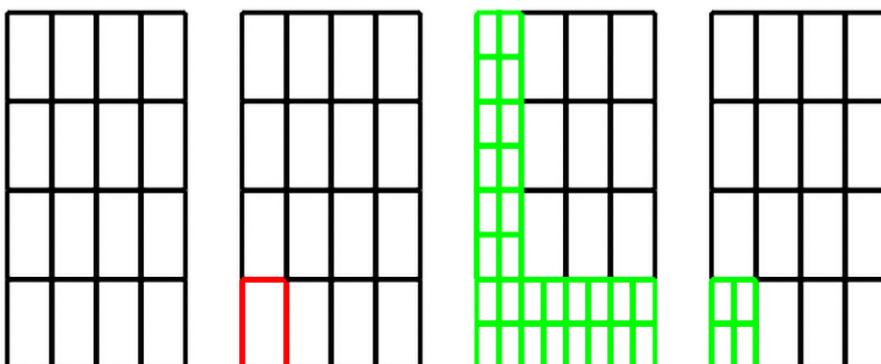


Fig. 5: Gaussian bell curve obtained by taking the tensor product of 2 B-spline functions

using the same notation as in the previous section. $(\otimes \mathbf{R}^k)$ is the tensor product of the univariate transfer matrices.

The drawback of this method (Forstley and Bartels 1988) is linked to the tensor product approach. Exemplarily, when inserting a knot in one direction, the corresponding univariate B-spline basis will be changed. Per construction, the refinement will propagate through the whole mesh (see Fig. 6). This effect is unwanted: (1) the number of CP to estimate will remain high, (2) this may lead to unwanted oscillations in the approximation and (3) we lack a local refinement *only*.

2.2.2 Knot insertion in 2D

The hierarchical setting

To face this challenge, hierarchical B-splines called Hb-splines were introduced (Forsey and Bartels 1988, 1995, Kraft 1998) and further improved by, e.g., Gianelli et al. (2002) by introducing the Truncated hierarchical B-splines (THB). The concept is similar to 1D case. Starting from an initial, tensor product mesh K^0 , some areas are selected for refinement. Then, several new meshlines are introduced, which halve the knot spans of the local

Fig. 6:
Tensor product drawback when performing knot insertion: starting from an initial mesh (left), the red cell is to be refined. Control points are inserted along the two directions without local control as initially needed (Fig. 6 right).

knot vectors of all the functions contained therein (dyadic refinement, see Fig. 7). Certain functions from the coarser domain are deleted and only active functions kept; certain functions of the finer domains are added. These latter build a subset of the corresponding tensor product functions (Johannessen et al. 2015).

Mathematically, the procedure of refinement has three steps:

- The first step is to initialize, i.e. to define the coarse mesh, which is called $K^0 = \{B \in B^0 : \text{supp } B \neq 0\}$, with $\text{supp } B = \{\mathbf{x} : B(\mathbf{x}) \neq 0 \wedge \mathbf{x} \in \Omega^0\}$, with \mathbf{x} the vector of the Euclidean coordinates of one point. In other words: in the initialization step all basis functions from the basis B^0 are selected, the support of which is completely contained in the domain Ω^0 . The support is the subset of the domain containing those elements which are not mapped to zero.
- The second step is to recursively construct the finer mesh.

$$\begin{aligned} K^{l+1} &= K_A^{l+1} \cup K_B^{l+1}, \quad l = 0 \dots n-2 \\ K_A^{l+1} &= \{B \in K^l : \text{supp } B \subset \Omega^{l+1}\}, \\ K_B^{l+1} &= \{B \in B^{l+1} : \text{supp } B \not\subset \Omega^{l+1}\} \end{aligned} \quad (10)$$

- The third step results in the hierarchical B-spline basis $K = K^{N-1}$.

The basis functions from $B^l, l = 0, \dots, N-1$ present in K are called active functions.

We note that the subdomains are considered to be nested, i.e. $\Omega^l \subseteq \Omega^{l-1}$, $\Omega = \{\Omega^l\}_{l=0 \dots N-1}$. By applying iterative dyadic refinement, the spline spaces are nested, as well as the knot vectors, which is an important property to derive mathematical properties of the hierarchical B-spline basis (see e.g. Kiss 2015).

We propose to illustrate the procedure by considering a simple case with a coarse and a fine mesh. Fig. 8 shows the corresponding tensor-product B-splines for the two levels. As previously, the green area is the one to be refined, *only*. The principle is similar to the 1D case, where only active functions of the coarser mesh are selected (K_A^{l+1}) and “glued” with the active function of the finer level (K_B^{l+1}). The result of the refined mesh can be visualized in Fig. 8 (bottom). The Gaussian bell curve structure comes from the equi-spaced knot vector.

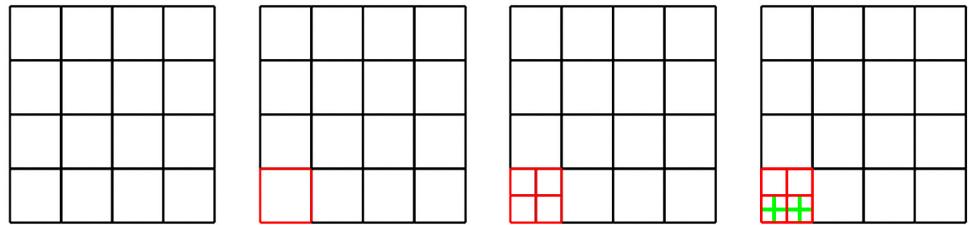


Fig. 7: Concept of mesh refinement. We start from a 4×4 mesh. In a first step, the red cell is refined dyadically and in a second refinement step, the two red cells on the bottom are refined a second time resulting in the green cells.

The hierarchical B-spline basis has nice properties such as local support, nonnegativity, piecewise structure, linear independence (Lyche et al. 2018). However, as in the 1D case, the partition of unity is lost and so the convex hull property mentioned in the 1D case, which is an important property in geometric design and is associated with some numerical conditioning.

$$\sum_{i=0}^{n^1} \sum_{j=0}^{n^2} B_{i,p^1,\mathbf{U}^1}(u^1) B_{j,p^2,\mathbf{U}^2}(u^2) \neq 1 \quad (11)$$

As shortly mentioned in the 1D case, the support of the coarse basis functions is large; they may overlap with the support of several fine-scale ones, leading to a “messy” base. This may affect negatively the solution of linear system as presented in the next section: more functions need to be evaluated leading to large but sparse matrices. In 2D, this drawback can become problematic. In this contribution, we will focus on Hb-splines and propose to consider further alternatives in a next contribution.

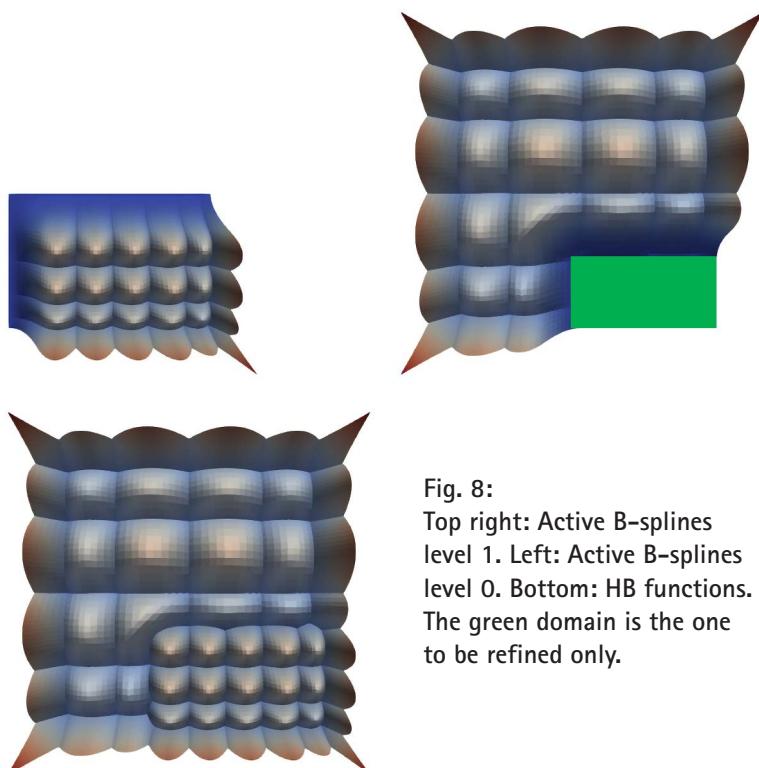


Fig. 8:
Top right: Active B-splines
level 1. Left: Active B-splines
level 0. Bottom: HB functions.
The green domain is the one
to be refined only.

3 Approximation of point clouds

3.1 Least-square approach

Up to now, we explained the construction of the refined mesh and the hierarchical B-spline basis. In this section, we address the approximation scheme of point clouds using a least-squares approach and introduce the refinement criterium. We start with a coarse mesh as defined in the previous section. At each iteration, the coefficients of the B-splines or control points are estimated by

$$\min_{p \in \mathbb{R}^n} \|Ap - I\|_\Sigma^2 \quad (12)$$

with $\|\cdot\|$ being the usual Euclidean distance. I is the vector of parametrized observations of size $[n_{obs}, 3]$; p is the matrix containing the coordinates of the control points of size $[n^1 n^2, 3]$.

On a regular mesh – without local refinement –, the elements of the design matrix A of size $[n_{obs}, n^1 n^2]$ contains the evaluation of the tensor product at the parameter values (Zhao et al. 2019). Σ is the variance covariance matrix (VCM) of the error term $v = Ap - I$, here assumed to be the identity matrix (homoscedastic observations).

Parametrized observations

The selection of parameters for every data point is the pre-step before performing surface approximation. Parameters are the unique numeric values of points on a surface and can be computed using, e.g., the uniform, chord length or centripetal method. All of these methods are derived from the spring model (Greiner and Hormann 1997), with the idea of replacing each edge of the data points by a spring. The boundary points are then projected into a plane and fixed there. The remaining vertices should arrange themselves so that the spring energy is minimized. Interested readers will find a detailed description of the different methods in Piegl and Tiller (1997). After having assigned parameter values $Q_i = (u_i^1, u_i^2)$ to every data point l_i , a rectangle $[a, b] \times [c, d]$ is chosen, that contains all parameter values.

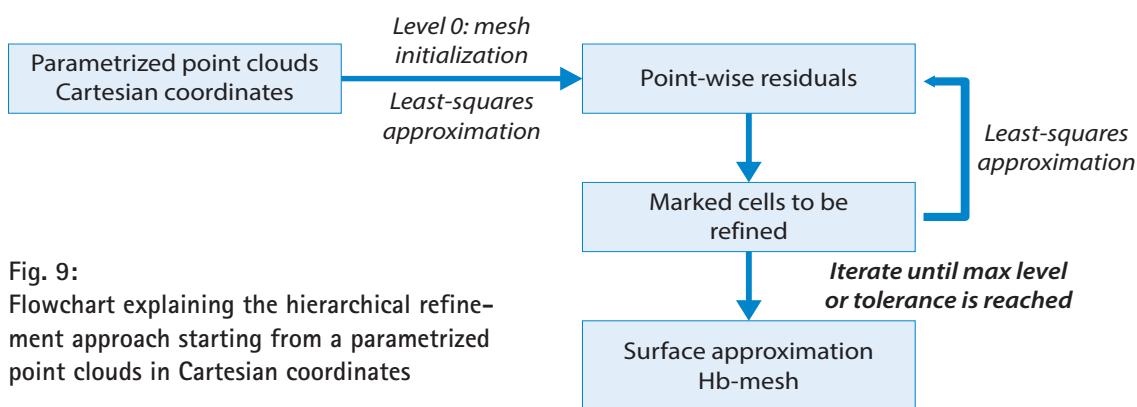


Fig. 9:
Flowchart explaining the hierarchical refinement approach starting from a parametrized point clouds in Cartesian coordinates

For an equidistant rectangular grid $\{(r_j, s_k)\}$, we have

$$r_j - r_{j-1} = \frac{b-a}{n_u^1}, \quad s_k - s_{k-1} = \frac{d-c}{n_u^2}.$$

3.2 Evaluation of the errors

The error term needs to be evaluated at each data points. The refinement is performed iteratively in the areas of the domain where the value of the root mean squared error (RMSE) exceeds a given tolerance ε . This procedure is repeated until the maximum number of iterations defined by the user is reached.

Consecutively, a new design matrix can be computed for each level following the procedure used for knot refinement. The matrix R allows to derive a new design matrix \bar{A} for the next level of refinement by selecting only the active functions of the coarser mesh and the active function of the finer levels. A detailed explanation of this procedure is provided in D'Angella et al. (2018).

At each refinement level, the CP have to be re-evaluated: the total number of CP to compute using Hb-splines is smaller than if each cell is refined (global approach). Thus, the number of B-splines to be evaluated is less important and the procedure computational efficient. The procedure is summarized in a flowchart form in Fig. 9. The next section uses an example to show how the Hb-splines allows to approximate a simulated point cloud.

3.3 Simulated TLS observations

In this section, we propose to highlight the advantages of the hierarchical B-spline approximation by comparing the results to the one obtained with the usual global approach.

To that aim, we simulate equally spaced data points corresponding to the test function

$$z = \left(\frac{3}{2} e^{\sqrt{(10x-3)^2 + (10y-3)^2}} \right)^{-1} + \left(\frac{3}{2} e^{\sqrt{(10x+3)^2 + (10y+3)^2}} \right)^{-1} + \left(\frac{3}{2} e^{\sqrt{(10x)^2 + (10y)^2}} \right)^{-1} \quad (13)$$

with $x \in [-1, 1]$, $y \in [-1, 1]$ expressed in [m]. We generated a total of 150 points in each direction.

To the z-coordinates of the point clouds is added a Gaussian noise having a standard deviation of $\sigma_0 = 0.001$ m generated with the `randn` function of Matlab. We consider the x and y coordinates as not affected by additional white noise. We perform the approximation using a tolerance of $\varepsilon = 1e^{-3}$ m, justified by the chosen noise level. A maximum of 5 iterations was set up for the refinement as well as the degree 2 for the B-splines.

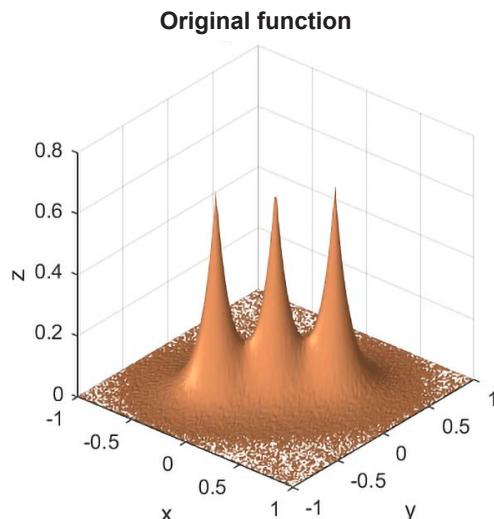
The original noisy surface as well as the corresponding approximation after the 5th iteration is shown in Fig. 10. The refined mesh of the last step is shown in Fig. 11 (right). For the sake of comparison, we add the mesh of the global approach in Fig. 11 (left); this latter is made of much more cells than the hierarchical mesh.

In Tab. 1, we present the average square error of the difference between the z-component of the reference

and the approximated surfaces estimated for each point at each iteration, $ase = \sqrt{\sum_{i=1}^{150*150} \frac{(S_{z,i} - z_i)^2}{150*150}}$ as well as the

maximum error $MAX = \max_i |(S_{z,i} - z_i)|$. We add further the number of cells of the meshes, the number of active functions that were evaluated, and the computation time needed to reach the 5th iteration. The same values were computed with the usual least-squares approach with the tensor-product of the B-splines of the 5th level. Please note that the computation time is only given as an indication and is subject to improvement; we are here interested in comparing *only* the differences between the approaches with and without refinement. The approximations were performed on a standard laptop.

Using the refined mesh (Fig. 11 right) at level 5, the *ave* reaches 0.001 m, and is, for the case under consideration, 5 times smaller compared to the value obtained with the



Hb approximation 5th level

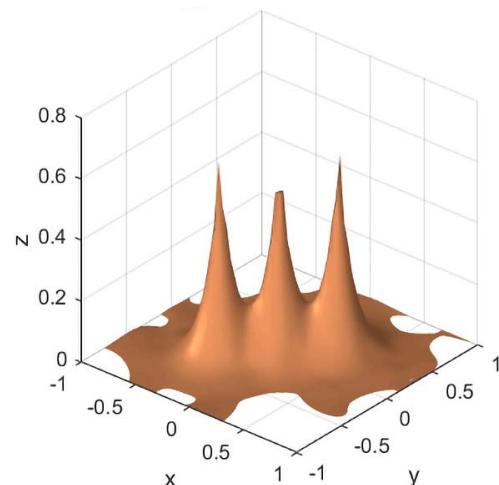


Fig. 10: Left: Original function as defined in Equation (13). A total number of 150 observations in the x- and y-axis were generated. The added Gaussian noise had a standard deviation of 0.001 m. Right: The approximation at the 5th iteration using hierarchical B-splines

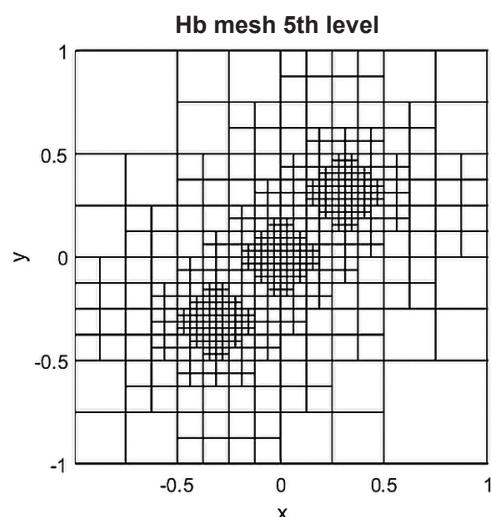
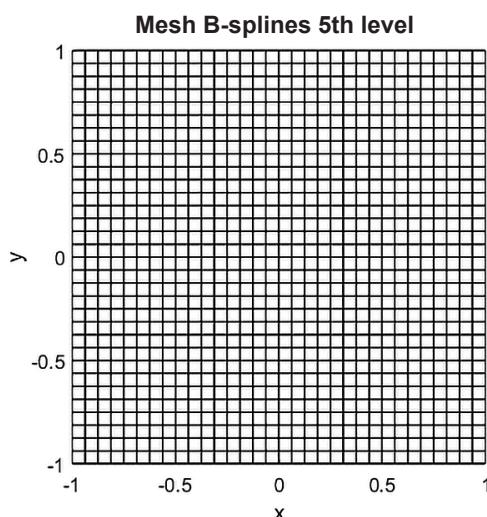


Fig. 11: Right: The global mesh at the 5th iteration. Left: The corresponding Hb-mesh

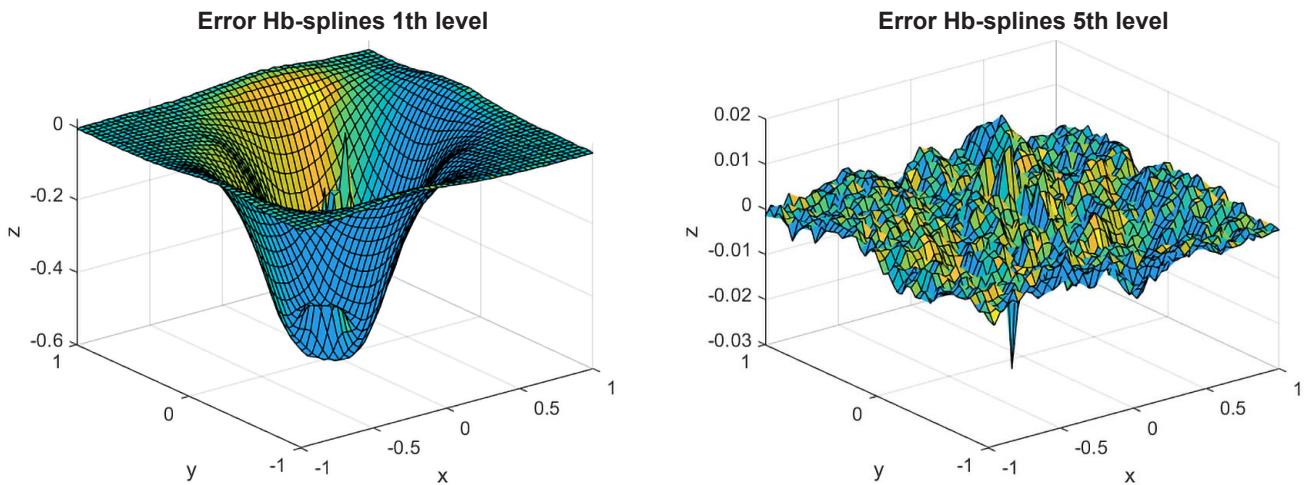


Fig. 12: Point-wise error between the approximation and the reference surface for the 1th (left) and 5th (right) level of refinement

Tab. 1: Comparison of the performance using the Hb-splines and the usual B-splines approach without refinement. A least-square solution was computed. The ave [m], max error [m], computation time [s] and number of active functions are given for each iteration.

	Hb-splines (Mesh in Fig. 11 left)	B-splines with- out refinement (Mesh in Fig. 11 right)
ave [m] Level 5	0.001	0.005
MAX [m] Level 5	0.031	0.150
Nb of B-splines Level 5	466	1156
Computational time to reach Level 5 [s]	4.47s	8.58s

non-refined mesh of the same level (Fig. 11 left). Even at level 4, the *ave* was found to be smaller by 0.002 m. Since the three peaks (Fig. 10) are badly approximated with the global approach (see section 2, Fig. 7), the maximum error reaches 0.15 m, 5 times higher than with the refined strategy. Clearly, the approximation of the 1th level is found to be coarse with an average error of 0.45 m.

These results highlight both the drawback of the tensor product approach as well as the great potential of the Hb-splines to approximate challenging geometries from noisy point clouds. At the 5th iteration, the number of parameters to estimate is decreased by a factor 2.5, from 1156 to 466 B-splines functions to evaluate. The computational time is slightly improved, the last iteration remaining highly time consuming. Improvement in that direction will be our next focus. Increasing the noise by an (unrealistic) factor 5 leaded to a smaller difference between hierarchical and global approach, the *ave* was only slightly decreased from 0.016 with the global approach to 0.013 with the Hb-splines. However, the maxi-

mum error was strongly reduced by a factor 2.5 with the hierarchical approach, highlighting the importance of local refinement for approximating the peaks of the original function.

4 Real case

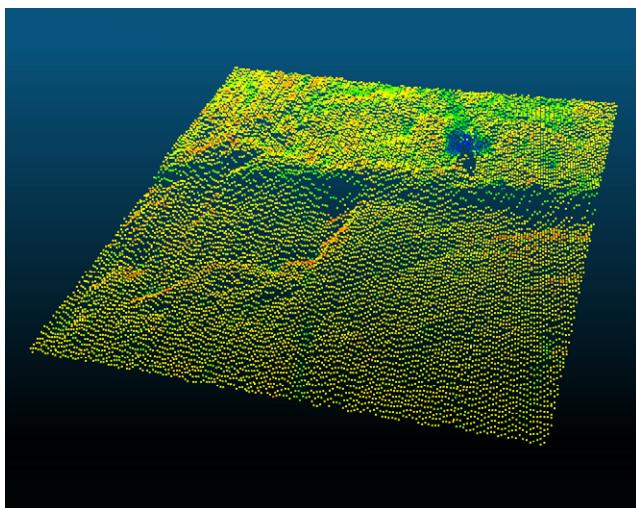
4.1 Description of the data set

In this section, we use a real data set of an historic masonry arch bridge over the river Aller near Verden in Germany to compare the approximations using the hierarchical B-splines and the global B-splines approach without adaptive refinement. Using a specific software (CloudCompare), a small surface with a challenging geometry was chosen.

4.2 Mathematical approximation

The obtained point cloud is approximated with a B-spline surface using a global and hierarchical approach, as described in Section 2. The parameterization is carried out using a uniform method, which is justified by the relatively smooth geometry of the PC. We chose an equidistant knot vector for the same reason. We used a tolerance of 4 mm to decide whether a cell should be further refined or not, which we justify by the expected noise level of the TLS PC using the intensity model (Wujanz et al. 2017). That way, the risk of noise fitting in the refined domains should be minimized. The result obtained with Hb-splines after the 4th iteration is depicted in Fig. 13 (right) together with a representation in CloudCompare of the original point cloud (Fig. 13 left).

The meshes of the two approaches after the 4th iteration are presented in Fig. 14: the Hb-spline surface fitting only refined cells where the RMSE exceeds the chosen



Approximated Pointcloud: 4-Iteration

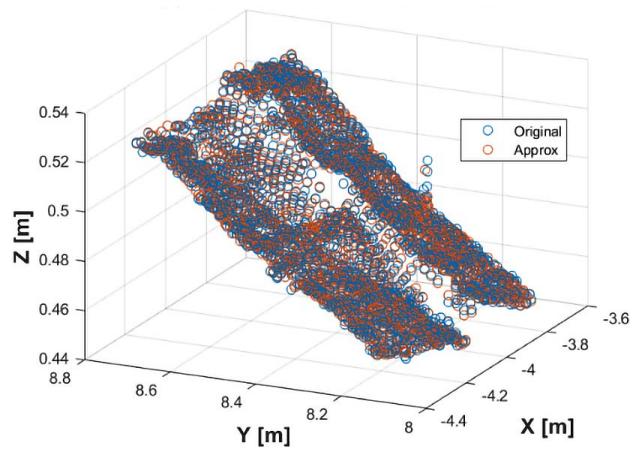
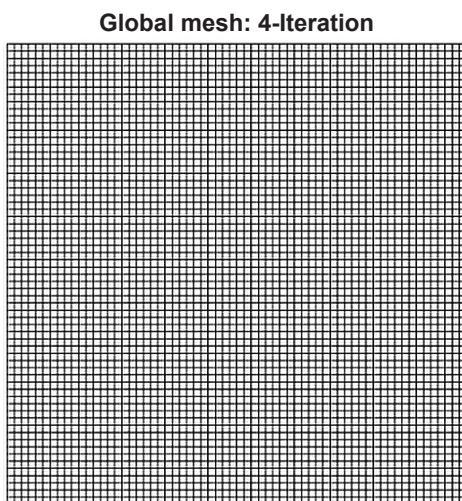
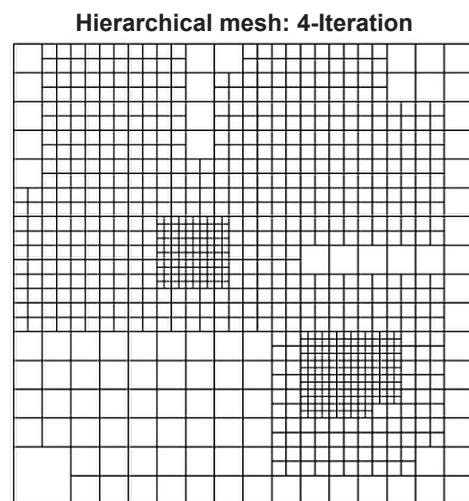


Fig. 13: Location of the chosen surface for approximation with the Hb-splines (red). Left: The surface under consideration shows edges, valleys and peaks. Right: The original point cloud (blue) and its B-spline surface approximation (red)



Global mesh: 4-Iteration



Hierarchical mesh: 4-Iteration

Fig. 14: Global mesh (left) and hierarchical mesh (right) after the 4th iteration

tolerance. Thus, only local details are approximated at each iteration whereas in the global approach, each cell will be refined.

The major advantage of the hierarchical refinement is further highlighted in Tab. 2. Whereas the number of CP exponentially increases at each iteration with a global refinement and reaches 4356 for the 4th iteration, only 898 parameters have to be estimated within the hierarchical framework. Using the same computer as for the simulations, 600 s were needed to compute the approximation at the 3rd iteration for the global approach compared to only 120 s for the hierarchical one. Due to the high computational time, the 4th iteration was not computed with the global approach. It confirms what the simulation already highlighted: the computational time decreases strongly when the Hb-spline are used with an adequate tolerance.

For that case study, the RMSE remained similar for both approaches (1.7 mm after the 3rd iteration), which is justified by the less challenging geometry as in the simulation.

Tab. 2: Number of CP to be estimated in the global and hierarchical approach. Root Mean Square Error (RMSE) and Maximum error are given in [mm].

Iteration	Parameter	Global	Hierarchical
1	Control points	100	100
	RMSE [mm]	2.4	2.4
	Max. Error [mm]	29.3	29.3
2	Control points	324	317
	RMSE [mm]	1.7	1.7
	Max. Error [mm]	24.1	24.1
3	Control points	1156	749
	RMSE [mm]	1.2	1.3
	Max. Error [mm]	22.2	22.2
4	Control points	4356	898
	RMSE [mm]	...	1.1
	Max. Error [mm]	...	18.4

5 Conclusion

Point clouds from scattered and noisy TLS observations can be approximated mathematically with regression B-splines surfaces. Such an approximation leads to a reduction of the PC and opens the door for the setting of rigorous testing strategies including accurate and realistic stochastic information of the PC. Usual surface fitting is based on a tensor product of B-spline functions and has following drawbacks:

- the high number of parameters to estimate may not lead to a high computational time and
- the use of a tensor product restricts local refinement, leading potentially to a noise fitting.

To face this challenge, we detailed didactically the concept of hierarchical B-splines by using for the first time a geodetic application: the fitting of a scattered and noisy TLS PC. Thanks to both simulated observations and real data, we showed that the definition of an adequate tolerance allows for a local refinement of the PC. We highlighted that both the number of parameters to estimate as well as the computational time involved decrease strongly with respect to the global approach. These results confirm that high quality approximations can be directly computed via hierarchical representations with the main advantage that a reduced number of degrees of freedom compared to the global refinement method is needed. Further analysis will focus on statistical testing with hierarchical B-spline approximation.

References

- Aigner, M., Jüttler, B. (2009): Distance Regression by Gauss-Newton-type Methods and Iteratively Re-weighted Least-Squares. *Computing*, Springer Wien, 86(2):73–87.
- Bourbaki, N. (1989): Elements of mathematics, Algebra I. Springer-Verlag. ISBN 3-540-64243-9.
- Bracco, C., Giannelli, C., Mazzia, F., Sestini, A. (2016): Bivariate hierarchical Hermite spline quasi-interpolation. *BIT* 56: 1165–1188.
- Bracco, C., Giannelli, C., Großmann, D., Sestini, A. (2017): Adaptive fitting with THB-splines: Error analysis and industrial applications. *Comput. Aided Geom. Design*, 62:239–252.
- Bureick, J., Alkhatib, H., Neumann, I. (2016): Robust spatial approximation of laser scanner points clouds by means of free-form curve approaches in deformation analysis. *J. Appl. Geod.*, 10:27–35.
- Cohen, E., Riesenfeld, R.F., Elber, G. (2001): Geometric Modeling with Splines: an Introduction. A K Peters/CRC Press, 638 Pages.
- Costantini, P. (2000): Curve and surface construction using variable degree polynomial splines. *Comput. Aided. Geom. Des.* 17, 419–446.
- D'Angella, D., Kollmannsberger, S., Rank, E., Reali, A. (2018): Multi-level Bézier extraction for hierarchical local refinement of Isogeometric Analysis. *Comput. Methods in Appl. Mech. Eng.*, 328:147–174.
- Davydov, O., Zeilfelder, F. (2004): Scattered data fitting by direct extension of local polynomials to bivariate splines. *Adv. Comp. Math.*, 21:223–271.
- de Boor, C.A. (1972): On calculating with B-splines. *J. Approx. Theory*, 6(1):50–62.
- de Boor, C.A. (2001): Practical Guide to Splines. Revised ed., Springer: New York, NY, USA.
- Forsey D.R., Bartels, R.H. (1988): Hierarchical B-spline refinement. *Comput. Graphics*, 22:205–212.
- Forsey, D.R., Bartels, R.H. (1995): Surface fitting with hierarchical splines. *ACM Trans. Graphics*, 14:134–161.
- Giannelli, C., Jüttler, B., Speleers, H. (2012): THB-splines: the truncated basis for hierarchical splines. *Comput. Aided Geom. Design*, 29, 485–498.
- Goshtasby, A., Cheng, F., Barsky, B.A. (1990): B-spline curves and surfaces viewed as digital filters. *Computer Vision, Graphics, and Image Processing*, 52(2):264–275.
- Greiner G., Hormann, K. (1997): Interpolating and approximating scattered 3D-data with hierarchical tensor product B-splines. In: Le Méhauté, A., Rabut, C., Schumaker, L.L. (eds.): *Surface Fitting and Multiresolution Methods*, 163–172. Vanderbilt University Press, Nashville, TN.
- Harmening, C., Neuner, H. (2015): A constraint-based parameterization technique for B-spline surfaces. *J. Appl. Geod.* 9:143–161.
- Harmening, C., Neuner, H. (2020): A spatio-temporal deformation model for laser scanning point clouds. *J. Geod.*, 94 (26).
- Kermarrec, G., Paffenholz, J.A., Alkhatib, H. (2019): How Significant Are Differences Obtained by Neglecting Correlations When Testing for Deformation: A Real Case Study Using Bootstrapping with Terrestrial Laser Scanner Observations Approximated by B-Spline Surfaces. *Sensors (Basel)*, 19(17):3640.
- Kiss, G. (2015): Theory and algorithms for truncated hierarchical B-splines. Phd 2015, OeBB (VLID)753073.
- Kjetil, A.J., Filippo, R., Trond, K. (2015): On the similarities and differences between Classical Hierarchical, Truncated Hierarchical and LR B-splines. *Comput.Methods in Appl.Mech.Eng.*, 291:64–101.
- Koch, K.R. (2009): Fitting free-form surfaces to laserscan data by NURBS. *AVN Allg. Vermess.-Nachr.* 116:134–140.
- Kraft, R. (1997): Adaptive and linearly independent multilevel B-splines. In: Le Méhauté, A., Rabut, C., Schumaker, L. L. (eds.): *Surface Fitting and Multiresolution Methods*, 209–218. Vanderbilt University Press, Nashville.
- Lee, I.K. (2000): Curve reconstruction from unorganized points. *Comput. Aided Geom.*, 17(2):161–177.
- Lyche, T., Manni, C., Speleers, H. (2018): Foundations of Spline Theory: B-Splines, Spline Approximation, and Hierarchical Refinement. In: Lyche, T., Manni, C., Speleers H. (eds.): *Splines and PDEs: From Approximation Theory to Numerical Linear Algebra*. Lecture Notes in Mathematics, vol 2219. Springer, Cham.
- Piegl, L.; Tiller, W. (1997): The NURBS Book. Springer Science & Business Media: Berlin, Germany.
- Rabut, C. (2005): Locally tensor product functions. *Numer. Algor.* 39, 329–348.
- Unser, M.A. (1997): Ten good reasons for using Spline Wavelets. In: Aldroubi, A., Laine, A.F., Unser, M.A. (eds.): *Wavelet Applications in Signal and Image Processing V*, 3169:422–431.
- Vuong, A.V., Gianelli, C., Büttler, B., Simeon, B. (2011): A hierarchical approach to adaptive local refinement in isogeometric analysis. *Comput. Methods in Appl. Mech. Eng.*, 200(49–52):3554–3567.
- Wujanz, D., Burger, M., Mettenleiter, M., Neitzel, F. (2017): An intensity-based stochastic model for terrestrial laser scanners. *ISPRS J. Photogramm. Remote Sens.*, 125, 146–155.

Contact

Gaël Kermarrec, kermarrec@gih.uni-hannover.de
 Knut Hartmann, hartmann-kn@gmx.net
 Jan Hartmann, jamo@ja-mo.eu
 Lixian Chen, chenlixian0910@gmail.com
 Hannes Faust, hannes.faust@hotmail.com
 Reza Besharat, besharatreza@gmail.com
 Gad Samuel, tellgad@gmail.com
 Hamza Alkhatib, alkhatib@gih.uni-hannover.de

Geodetic Institute, Leibniz Universität Hannover
 Nienburger Straße 1, 30167 Hannover, Germany

This article also is digitally available under www.geodesie.info.